# PSD913F2
# PSD934F2 PSD954F2

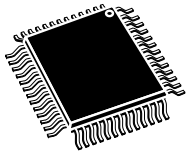
## Flash In-System Programmable (ISP) Peripherals For 8-bit MCUs

**PRELIMINARY DATA**

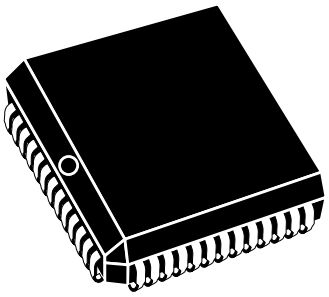
### FEATURES SUMMARY

- Single Supply Voltage:
  - 5 V±10% for PSD9xxF2
  - 3.3 V±10% for PSD9xxF2-V
- Up to 2Mbit of Primary Flash Memory (8 uniform sectors)
- 256Kbit Secondary Flash Memory (4 uniform sectors)
- Up to 256Kbit SRAM
- Over 2,000 Gates of PLD: DPLD
- 27 Reconfigurable I/O ports
- Enhanced JTAG Serial Port
- Programmable power management
- High Endurance:
  - 100,000 Erase/Write Cycles of Flash Memory
  - 1,000 Erase/Write Cycles of PLD

**Figure 1. Packages**



PQFP52 (T)



PLCC52 (K)



This is preliminary information on a new product now in development or undergoing evaluation. Details are subject to change without notice.

# PSD9XX Family

## PSD913F2 PSD934F2 PSD954F2

## Configurable Memory System on a Chip for 8-Bit Microcontrollers

### Table of Contents


Introduction....1
- In-System Programming (ISP) JTAG....2
- In-Application Programming (IAP)....2

Key Features....3
Block Diagram....4
PSD9XX Family....5
Architectural Overview....6
- Memory....6
- Page Register....6
- PLDs....6
- I/O Ports....7
- Microcontroller Bus Interface....7
- JTAG Port....7
- In-System Programming....8
- Power Management Unit....8

Development System....9
Pin Descriptions....10
Register Description and Address Offset....14
Functional Blocks....15
- Memory Blocks....15
  - Main Flash and Secondary Flash Memory Description....15
  - SRAM....27
  - Memory Chip Selects....27
  - Page Register....30
- PLDs....31
  - Decode PLD (DPLD)....33
  - General Purpose PLD (GPLD)....33
- Microcontroller Bus Interface....35
  - Interface to a Multiplexed 8-bit Bus....35
  - Interface to a Non-multiplexed 8-bit Bus....35
  - Microcontroller Interface Examples....37
- I/O Ports....42
  - General Port Architecture....42
  - Port Operating Modes....44
  - Port Configuration Registers (PCRs)....47
  - Port Data Registers....49
  - Ports A and B – Functionality and Structure....49
  - Port C – Functionality and Structure....51
  - Port D – Functionality and Structure....51



***For additional information,***
***Call 800-832-6974***
***Fax: 510-657-8495***
***Web Site: http://www.psdst.com***
***E-mail: ask.psd@st.com***

### Table of Contents


Power Management....54
Automatic Power Down (APD) Unit and Power Down Mode....54
Other Power Savings Options....58
Reset and Power On Requirement....59
Programming In-Circuit using the JTAG Interface....60
Standard JTAG Signals....61
JTAG Extensions....61
Security and Flash Memories Protection....61
Absolute Maximum Ratings....62
Operating Range....62
Recommended Operating Conditions....62
AC/DC Parameters....63
Example of Typical Power Calculation at Vcc = 5..0 V....64
Example of Typical Power Calculation at Vcc = 5..0 V in Turbo Off Mode....65
DC Characteristics (5 V ± 10% versions)....66
Microcontroller Interface – AC/DC Parameters (5 V ± 10% versions)....67
Read Timing....68
Write Timing....69
PLD Combinatorial Timing....69
Power Down Timing....70
Vstbyon Timing....70
Reset Pin Timing....70
Flash Program, Write and Erase Times....71
ISC Timing....71
PSD9XXFV DC Characteristics (3.0 V to 3.6 V Versions) Advance Information....72
Microcontroller Interface – AC/DC Parameters (3 V versions)....73
Read Timing (3 V versions)....73
Write Timing (3 V versions)....74
PLD Combinatorial Timing (3 V versions)....74
Power Down Timing (3 V Versions)....75
$V_{stbyon}$ Timing (3 V Versions)....75
Reset Pin Timing (3 V Versions)....75
Flash Program, Write and Erase Times (3 V Versions)....76
ISC Timing (3 V Versions)....76



***For additional information,***
***Call 800-832-6974***
***Fax: 510-657-8495***
***Web Site: http://www.psdst.com***
***E-mail: ask.psd@st.com***

# PSD9XX Family

## PSD913F2 PSD934F2 PSD954F2

## Configurable Memory System on a Chip for 8-Bit Microcontrollers

### Table of Contents


Timing Diagrams ....77
Pin Capacitance ....81
AC Testing Input/Output Waveforms....81
AC Testing Load Circuit....81
Programming....81
Pin Assignments....82
Package Information....84
Selector Guide....87
Part Number Construction....87
Ordering Information....88
Document Revisions....89



**For additional information,**
**Call 800-832-6974**
**Fax: 510-657-8495**
**Web Site: http://www.psdst.com**
**E-mail: ask.psd@st.com**

# PSD913F2, PSD934F2, PSD954F2

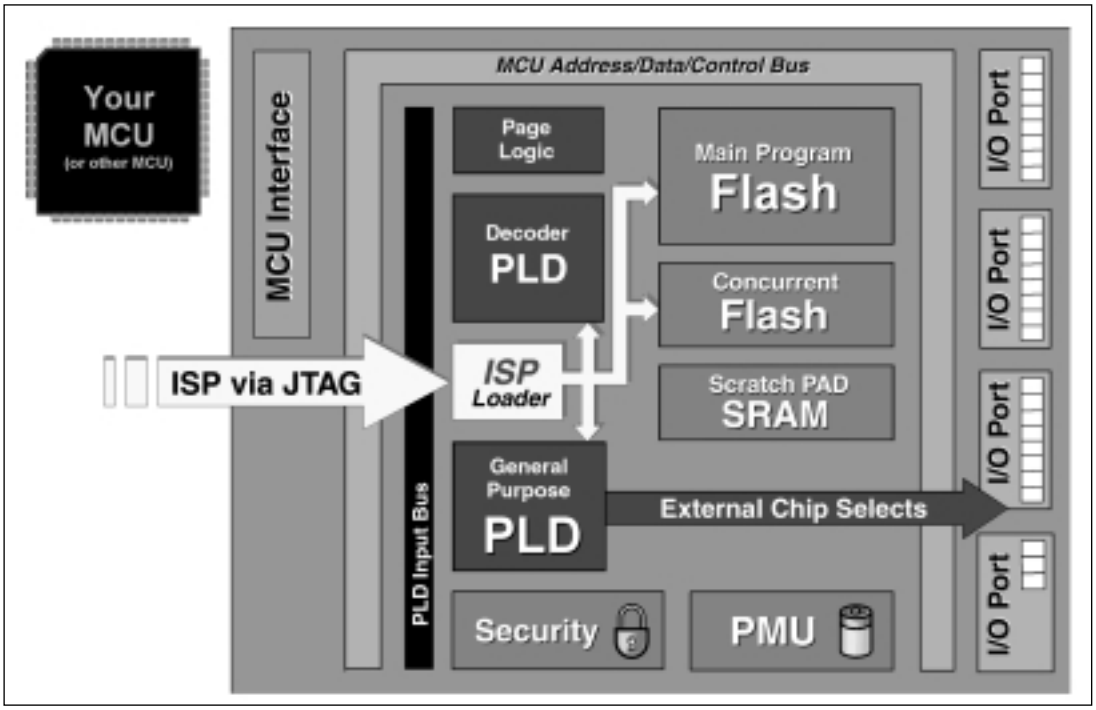
## Configurable Memory System on a Chip for 8-Bit Microcontrollers

Preliminary Information

### 1.0 Introduction

The PSD9XX family of Programmable System Devices (for 8-bit microcontrollers) brings In-System-Programmability (ISP) to Flash memory and programmable logic. The result is a simple and flexible solution for embedded designs. PSD9XX devices combine many of the peripheral functions found in MCU based applications:

- Up to 2 Mbit of Flash memory
- A secondary 256 Kbit Flash memory
- Over 2,000 gates of Flash programmable logic
- Up to 256 Kbit SRAM
- Reconfigurable I/O ports
- Programmable power management.



## 1.0 Introduction (Cont.)

The PSD9XX family offers two methods to program PSD Flash memory while the PSD is soldered to a circuit board.

❑ In-System Programming (ISP) JTAG
An IEEE 1149.1 compliant JTAG interface is included on the PSD enabling the entire device (both flash memories, the PLD, and all configuration) to be rapidly programmed while soldered to the circuit board. This requires no MCU participation, which means the PSD can be programmed anytime, even while completely blank.

The innovative JTAG interface to flash memories is an industry first, solving key problems faced by designers and manufacturing houses, such as:

- **First time programming –** How do I get firmware into the flash the very first time? JTAG is the answer, program the PSD while blank with no MCU involvement.
- **Inventory build-up of pre-programmed devices –** How do I maintain an accurate count of pre-programmed flash memory and PLD devices based on customer demand? How many and what version? JTAG is the answer, build your hardware with blank PSDs soldered directly to the board and then custom program just before they are shipped to customer. No more labels on chips and no more wasted inventory.
- **Expensive sockets –** How do I eliminate the need for expensive and unreliable sockets? JTAG is the answer. Solder the PSD directly to the circuit board. Program first time and subsequent times with JTAG. No need to handle devices and bend the fragile leads.

❑ In-Application Programming (IAP)
Two independent flash memory arrays are included so the MCU can execute code from one memory while erasing and programming the other. Robust product firmware updates in the field are possible over any communication channel (CAN, Ethernet, UART, J1850, etc) using this unique architecture. Designers are relieved of these problems:

- **Simultaneous read and write to flash memory –** How can the MCU program the same memory from which it is executing code? It cannot. The PSD allows the MCU to operate the two flash memories concurrently, reading code from one while erasing and programming the other during IAP.
- **Complex memory mapping –** I have only a 64K-byte address space to start with. How can I map these two memories efficiently? A Programmable Decode PLD is the answer. The concurrent PSD memories can be mapped anywhere in MCU address space, segment by segment with extremely high address resolution. As an option, the secondary flash memory can be swapped out of the system memory map when IAP is complete. A built-in page register breaks the 64K-byte address limit.
- **Separate program and data space –** How can I write to flash memory while it resides in “program” space during field firmware updates, my MCU won’t allow it! The flash PSD provides means to “reclassify” flash memory as “data” space during IAP, then back to “program” space when complete.

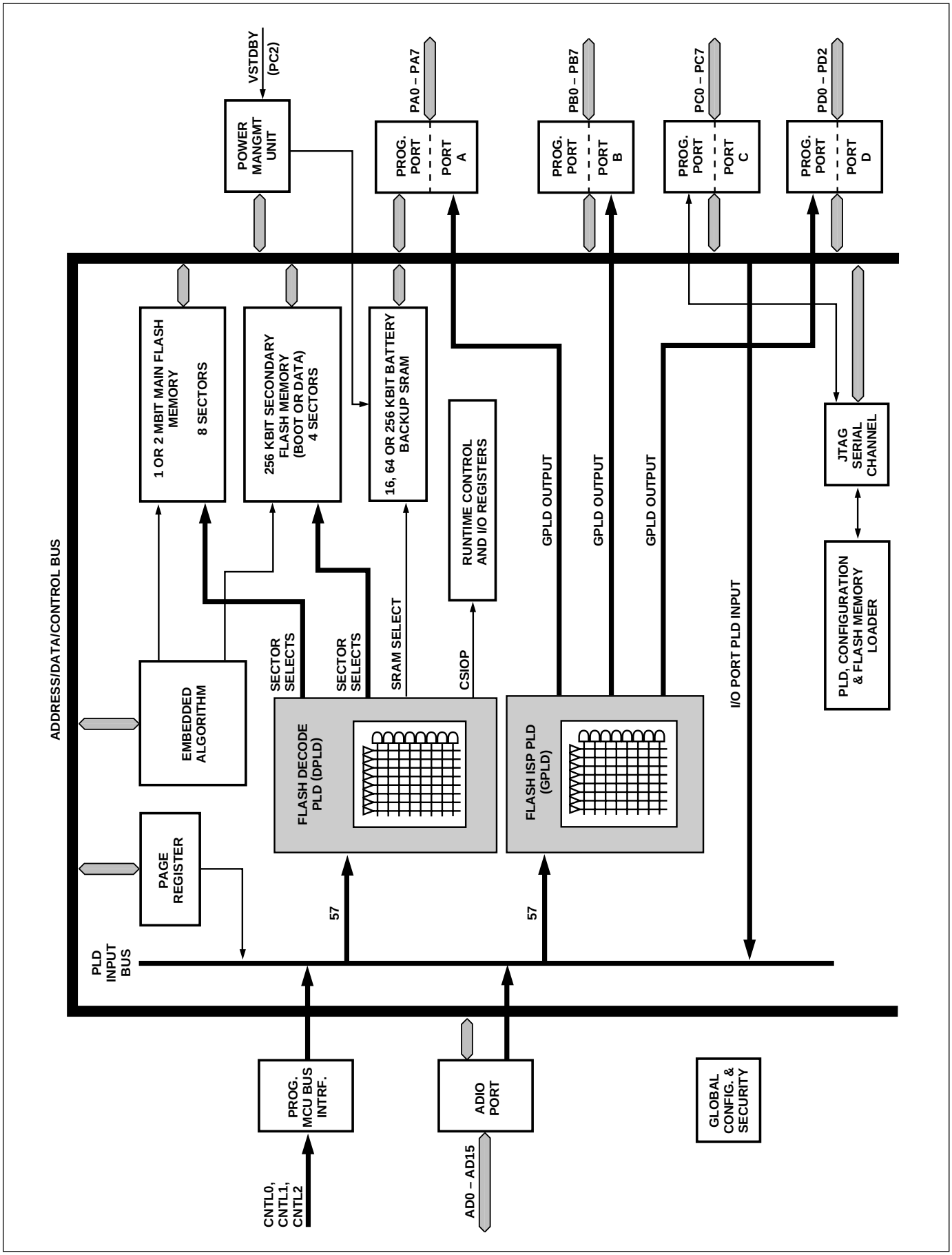
PSDsoft Express – ST’s software development tool – guides you through the design process step-by-step making it possible to complete an embedded MCU design capable of ISP/IAP in just hours. Select your MCU and PSDsoft Express will take you through the remainder of the design with point and click entry, covering...PSD selection, pin definitions, programmable logic inputs and outputs, MCU memory map definition, ANSI C code generation for your MCU, and merging your MCU firmware with the PSD design. When complete, two different device programmers are supported directly from PSDsoft – FlashLINK (JTAG) and PSDpro.

The PSD9XX is available in 52-pin PLCC and PQFP packages as well as a 64-pin TQFP package.

## 2.0 Key Features

- A simple interface to 8-bit microcontrollers that use either multiplexed or non-multiplexed busses. The bus interface logic uses the control signals generated by the microcontroller automatically when the address is decoded and a read or write is performed. A partial list of the MCU families supported include:
  - Intel 8031, 80196, 80186, 80C251
  - Motorola 68HC11, 68HC16, 68HC12, and 683XX
  - Philips 8031 and 8051XA
  - Zilog Z80, Z8, and Z180
- Internal 1 or 2 Mbit flash memory. This is the main Flash memory. It is divided into eight equal-sized blocks that can be accessed with user-specified addresses.
- Internal secondary 256 Kbit Flash memory. It is divided into four equal-sized blocks that can be accessed with user-specified addresses. This secondary memory brings the ability to execute code and update the main Flash **concurrently.**
- 16, 64 or 256 Kbit SRAM. The SRAM's contents can be protected from a power failure by connecting an external battery.
- General Purpose PLD (GPLD) with 19 outputs. The GPLD may be used to implement external chip selects or combinatorial logic function.
- Decode PLD (DPLD) that decodes address for selection of internal memory blocks.
- 27 individually configurable I/O port pins that can be used for the following functions:
  - MCU I/Os
  - PLD I/Os
  - Latched MCU address output
  - Special function I/Os.
  - 16 of the I/O ports may be configured as open-drain outputs.
- Standby current as low as 50 µA for 5 V devices.
- Built-in JTAG compliant serial port allows full-chip In-System Programmability (ISP). With it, you can program a blank device or reprogram a device in the factory or the field.
- Internal page register that can be used to expand the microcontroller address space by a factor of 256.
- Internal programmable Power Management Unit (PMU) that supports a low power mode called Power Down Mode. The PMU can automatically detect a lack of microcontroller activity and put the PSD9XX into Power Down Mode.
- Erase/Write cycles:
  - Flash memory – 100,000 minimum
  - PLD – 1,000 minimum
  - Data Retention: 15 years

Figure 1. PSD9XX Block Diagram



## 4.0 PSD9XX Family

There are 2 variants in the PSD9XX family. All PSD9XX devices provide these base features: 1 or 2 Mbit main Flash Memory, JTAG port, GPLD, DPLD, power management, and 27 I/O pins. The following table summarizes all the devices in the PSD9XX family. Additional devices will be introduced.

Table 1. PSD9XX Product Matrix

| Part # | | I/O Pins | No. of GPLD Output | Serial ISP JTAG/ISC Port | Flash Main Memory Kbit (8 Sectors) | Secondary Flash Memory Kbit (4 Sectors) | SRAM Kbit | Turbo Mode | Supply Voltage |
|---|---|---|---|---|---|---|---|---|---|
| PSD9XX Family | Device | | | | | | | | |
| PSD9XX | PSD913F2 | 27 | 19 | Yes | 1024 | 256 | 16 | Yes | 3V/5V |
| | PSD934F2 | 27 | 19 | Yes | 2048 | 256 | 64 | Yes | 3V/5V |
| | PSD954F2 | 27 | 19 | Yes | 2048 | 256 | 256 | Yes | 3V/5V |

# 5.0 PSD9XX Architectural Overview

PSD9XX devices contain several major functional blocks. Figure 1 shows the architecture of the PSD9XX device family. The functions of each block are described briefly in the following sections. Many of the blocks perform multiple functions and are user configurable.

## 5.1 Memory

The PSD9XX contains the following memories:

- A 1 or 2 Mbit Flash
- A secondary 256 Kbit Flash memory
- 16, 64 or 256 Kbit SRAM.

Each of the memories is briefly discussed in the following paragraphs. A more detailed discussion can be found in section 9.

The 1 or 2 Mbit Flash is the main memory of the PSD9XX. It is divided into eight equally-sized sectors that are individually selectable.

The 256 Kbit secondary Flash memory is divided into four equally-sized sectors. Each sector is individually selectable. This memory can hold boot code or data.

The SRAM is intended for use as a scratchpad memory or as an extension to the microcontroller SRAM. If an external battery is connected to the PSD9XX's Vstby pin, data will be retained in the event of a power failure.

Each block of memory can be located in a different address space as defined by the user. The access times for all memory types includes the address latching and DPLD decoding time.

## 5.2 Page Register

The eight-bit Page Register expands the address range of the microcontroller by up to 256 times.The paged address can be used as part of the address space to access external memory and peripherals or internal memory and I/O. The Page Register can also be used to change the address mapping of blocks of Flash memory into different memory spaces IAP.

## 5.3 PLDs

The device contains two combinatorial PLD blocks, each optimized for a different function, as shown in Table 2. The functional partitioning of the PLDs reduces power consumption, optimizes cost/performance, and eases design entry.

The Decode PLD (DPLD) is used to decode addresses and generate chip selects for the PSD9XX internal memory and registers. The General Purpose PLD (GPLD) can implement user-defined external chip selects and logic functions. The PLDs receive their inputs from the PLD Input Bus and are differentiated by their output destinations, number of Product Terms.

The PLDs consume minimal power by using Zero-Power design techniques. The speed and power consumption of the PLD is controlled by the Turbo Bit in the PMMR0 register and other bits in the PMMR2 registers. These registers are set by the microcontroller at runtime. There is a slight penalty to PLD propagation time when invoking the non-Turbo bit.

Table 2. PLD I/O Table

| Name | Abbreviation | Inputs | Outputs | Product Terms |
|---|---|---|---|---|
| Decode PLD | DPLD | 57 | 15 | 39 |
| General Purpose PLD | GPLD | 57 | 19 | 114 |

PSD9XX Architectural Overview (cont.)

## 5.4 I/O Ports

The PSD9XX has 27 I/O pins divided among four ports (Port A, B, C, and D). Each I/O pin can be individually configured for different functions. Ports A, B, C and D can be configured as standard MCU I/O ports, PLD I/O, or latched address outputs for microcontrollers using multiplexed address/data busses.

The JTAG pins can be enabled on Port C for In-System Programming (ISP).

Port A can also be configured as a data port for a non-multiplexed bus.

## 5.5 Microcontroller Bus Interface

The PSD9XX easily interfaces with most 8-bit microcontrollers that have either multiplexed or non-multiplexed address/data busses. The device is configured to respond to the microcontroller's control signals, which are also used as inputs to the PLDs. Section 9.3.5 contains microcontroller interface examples.

## 5.6 JTAG Port

In-System Programming can be performed through the JTAG pins on Port C. This serial interface allows complete programming of the entire PSD9XX device. A blank device can be completely programmed. The JTAG signals (TMS, TCK, $\overline{\text{TSTAT}}$, $\overline{\text{TERR}}$, TDI, TDO) are enabled on Port C when selected or when a device is blank. Table 3 indicates the JTAG signals pin assignments.

Table 3. JTAG Signals on Port C

| Port C Pins | JTAG Signal |
|---|---|
| PC0 | TMS |
| PC1 | TCK |
| PC3 | $\overline{\text{TSTAT}}$ |
| PC4 | $\overline{\text{TERR}}$ |
| PC5 | TDI |
| PC6 | TDO |

PSD9XX Architectural Overview (cont.)

## 5.7 In-System Programming

Using the JTAG signals on Port C, the entire PSD9XX device can be programmed or erased without the use of the microcontroller (ISP). The main Flash memory can also be programmed in-system by the microcontroller executing the programming algorithms out of the Secondary Flash memory, or SRAM (IAP). The Secondary Flash memory can be programmed the same way by executing out of the main Flash memory. The PLD logic or other PSD9XX configuration can be programmed through the JTAG port or a device programmer. Table 4 indicates which programming methods can program different functional blocks of the PSD9XX.

Table 4. Methods of Programming Different Functional Blocks of the PSD9XX

| Functional Block | JTAG-ISP | Device Programmer | IAP |
|---|---|---|---|
| Main Flash Memory | Yes | Yes | Yes |
| Secondary Flash Memory | Yes | Yes | Yes |
| PLD Array (DPLD and GPLD) | Yes | Yes | No |
| PSD Configuration | Yes | Yes | No |

## 5.8 Power Management Unit

The Power Management Unit (PMU) in the PSD9XX gives the user control of the power consumption on selected functional blocks based on system requirements. The PMU includes an Automatic Power Down unit (APD) that will turn off device functions due to microcontroller inactivity. The APD unit has a Power Down Mode that helps reduce power consumption.

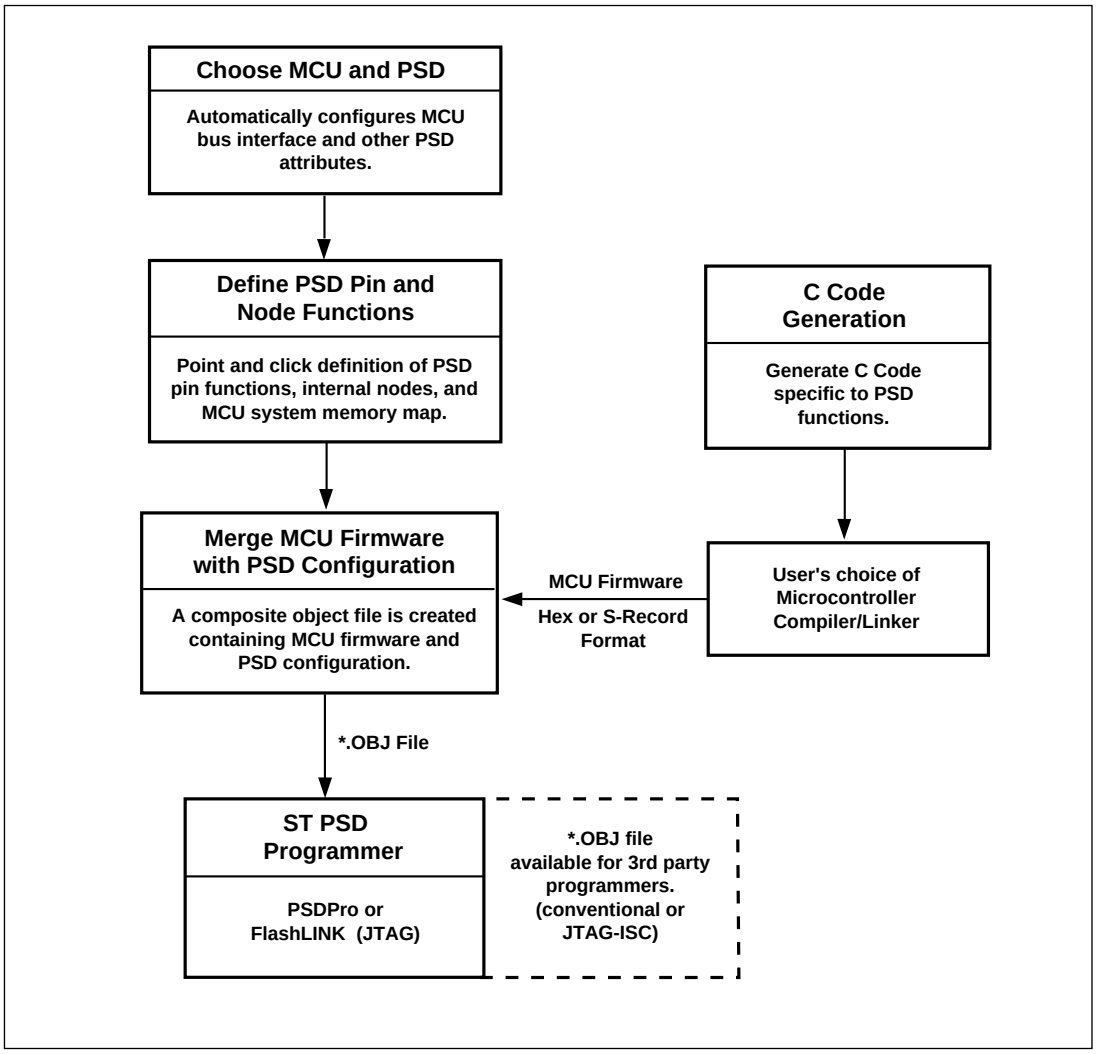
The PSD9XX also has some bits that are configured at run-time by the MCU to reduce power consumption of the PLD. The turbo bit in the PMMR0 register can be turned off and the PLD will latch its outputs and go to sleep until the next transition on its inputs. Additionally, bits in the PMMR2 register can be set by the MCU to block signals from entering the PLD to reduce power consumption. See section 9.5.

## 6.0 Development System

The PSD9XX family is supported by PSDsoft Express, a Windows-based (95, 98, NT) software development tool. A PSD design is quickly and easily produced in a point and click environment. The designer does not need to enter Hardware Definition Language (HDL) equations to define PSD pin functions and memory map information. The general design flow is shown in Figure 2 below. PSDsoft Express is available free from our web site (www.psdst.com) or the Literature CD.

PSDsoft Express directly supports two low cost device programmers from ST, PSDpro and FlashLINK (JTAG). Both of these programmers may be purchased through your local rep/distributor, or directly from our web site using a credit card. The PSD9XX is also supported by third party device programmers, see web site for current list.

Figure 2. PSDsoft Development Tools


Choose MCU and PSD
Automatically configures MCU bus interface and other PSD attributes.
Define PSD Pin and Node Functions
Point and click definition of PSD pin functions, internal nodes, and MCU system memory map.
C Code Generation
Generate C Code specific to PSD functions.
Merge MCU Firmware with PSD Configuration
A composite object file is created containing MCU firmware and PSD configuration.
MCU Firmware
Hex or S-Record Format
User's choice of Microcontroller Compiler/Linker
*.OBJ File
ST PSD Programmer
PSDPro or FlashLINK (JTAG)
*.OBJ file available for 3rd party programmers. (conventional or JTAG-ISC)

## 7.0 Table 5. PSD9XX Pin Descriptions

The following table describes the pin names and pin functions of the PSD9XX. Pins that have multiple names and/or functions are defined using PSDsoft.

| Pin Name | Pin* (PLCC) | Type | Description |
|---|---|---|---|
| ADIO0-7 | 30-37 | I/O | This is the lower Address/Data port. Connect your MCU address or address/data bus according to the following rules:<br>1. If your MCU has a multiplexed address/data bus where the data is multiplexed with the lower address bits, connect AD[0:7] to this port.<br>2. If your MCU does not have a multiplexed address/data bus, or you are using an 80C251 in page mode, connect A[0:7] to this port.<br>3. If you are using an 80C51XA in burst mode, connect A4/D0 through A11/D7 to this port.<br>ALE or AS latches the address. The PSD drives data out only if the read signal is active and one of the PSD functional blocks was selected. The addresses on this port are passed to the PLDs. |
| ADIO8-15 | 39-46 | I/O | This is the upper Address/Data port. Connect your MCU address or address/data bus according to the following rules:<br>1. If your MCU has a multiplexed address/data bus where the data is multiplexed with the lower address bits, connect A[8:15] to this port.<br>2. If your MCU does not have a multiplexed address/data bus, connect A[8:15] to this port.<br>3. If you are using an 80C251 in page mode, connect AD[8:15] to this port.<br>4. If you are using an 80C51XA in burst mode, connect A12/D8 through A19/D15 to this port.<br>ALE or AS latches the address. The PSD drives data out only if the read signal is active and one of the PSD functional blocks was selected. The addresses on this port are passed to the PLDs. |
| CNTL0 | 47 | I | The following control signals can be connected to this port, based on your MCU:<br>1. $\overline{WR}$ — active-low write input.<br>2. R_$\overline{W}$ — active-high read/active low write input.<br>This pin is connected to the PLDs. Therefore, these signals can be used in decode and other logic equations. |
| CNTL1 | 50 | I | The following control signals can be connected to this port, based on your MCU:<br>1. $\overline{RD}$ — active-low read input.<br>2. E — E clock input.<br>3. $\overline{DS}$ — active-low data strobe input.<br>4. $\overline{PSEN}$ — connect PSEN to this port when it is being used as an active-low read signal. For example, when the 80C251 outputs more than 16 address bits, PSEN is actually the read signal.<br>This pin is connected to the PLDs. Therefore, these signals can be used in decode and other logic equations. |

Table 5. PSD9XX Pin Descriptions (cont.)

| Pin Name | Pin* (PLCC) | Type | Description |
|---|---|---|---|
| CNTL2 | 49 | I | This pin can be used to input the $\overline{\text{PSEN}}$ (Program Select Enable) signal from any MCU that uses this signal for code exclusively. If your MCU does not output a Program Select Enable signal, this port can be used as a generic input. This port is connected to the PLDs. |
| $\overline{\text{Reset}}$ | 48 | I | Active low reset input. Resets I/O Ports and some of the configuration registers. Must be active at power up. |
| PA0<br>PA1<br>PA2<br>PA3<br>PA4<br>PA5<br>PA6<br>PA7 | 29<br>28<br>27<br>25<br>24<br>23<br>22<br>21 | I/O | These pins make up Port A. These port pins are configurable and can have the following functions:<br>1. MCU I/O — write to or read from a standard output or input port.<br>2. General Purpose PLD outputs.<br>3. Inputs to the PLDs.<br>4. Latched address outputs (see Table 6).<br>5. Address inputs. For example, PA0-3 could be used for A[0:3] when using an 80C51XA in burst mode.<br>6. As the data bus inputs D[0:7] for non-multiplexed address/data bus MCUs.<br>7. D0/A16-D3/A19 in M37702M2 mode.<br>**Note:** PA0-3 can only output CMOS signals with an option for high slew rate. However, PA4-7 can be configured as CMOS or Open Drain Outputs. |
| PB0<br>PB1<br>PB2<br>PB3<br>PB4<br>PB5<br>PB6<br>PB7 | 7<br>6<br>5<br>4<br>3<br>2<br>52<br>51 | I/O | These pins make up Port B. These port pins are configurable and can have the following functions:<br>1. MCU I/O — write to or read from a standard output or input port.<br>2. General Purpose PLD outputs.<br>3. Inputs to the PLDs.<br>4. Latched address outputs (see Table 6).<br>**Note:** PB0-3 can only output CMOS signals with an option for high slew rate. However, PB4-7 can be configured as CMOS or Open Drain Outputs. |
| PC0 | 20 | I/O | PC0 pin of Port C. This port pin can be configured to have the following functions:<br>1. MCU I/O — write to or read from a standard output or input port.<br>2. Input to the PLDs.<br>3. TMS Input for the JTAG Interface.<br>This pin can be configured as a CMOS or Open Drain output. |
| PC1 | 19 | I/O | PC1 pin of Port C. This port pin can be configured to have the following functions:<br>1. MCU I/O — write to or read from a standard output or input port.<br>2. Input to the PLDs.<br>3. TCK Input for the JTAG Interface.<br>This pin can be configured as a CMOS or Open Drain output. |

Table 5. PSD9XX Pin Descriptions (cont.)

| Pin Name | Pin* (PLCC) | Type | Description |
|---|---|---|---|
| PC2 | 18 | I/O | PC2 pin of Port C. This port pin can be configured to have the following functions:<br>1. MCU I/O — write to or read from a standard output or input port.<br>2. Input to the PLDs.<br>3. Vstby — SRAM standby voltage input for SRAM battery backup.<br>This pin can be configured as a CMOS or Open Drain output. |
| PC3 | 17 | I/O | PC3 pin of Port C. This port pin can be configured to have the following functions:<br>1. MCU I/O — write to or read from a standard output or input port.<br>2. Input to the PLDs.<br>3. $\overline{\text{TSTAT}}$ output for the JTAG interface.<br>4. Rdy/Bsy output for in-system parallel programming.<br>This pin can be configured as a CMOS or Open Drain output. |
| PC4 | 14 | I/O | PC4 pin of Port C. This port pin can be configured to have the following functions:<br>1. MCU I/O — write to or read from a standard output or input port.<br>2. Input to the PLDs.<br>3. $\overline{\text{TERR}}$ output for the JTAG interface.<br>4. Vbaton — battery backup indicator output. Goes high when power is being drawn from an external battery.<br>This pin can be configured as a CMOS or Open Drain output. |
| PC5 | 13 | I/O | PC5 pin of Port C. This port pin can be configured to have the following functions:<br>1. MCU I/O — write to or read from a standard output or input port.<br>2. Input to the PLDs.<br>3. TDI input for the JTAG interface.<br>This pin can be configured as a CMOS or Open Drain output. |
| PC6 | 12 | I/O | PC6 pin of Port C. This port pin can be configured to have the following functions:<br>1. MCU I/O — write to or read from a standard output or input port.<br>2. Input to the PLDs.<br>3. TDO output for the JTAG interface.<br>This pin can be configured as a CMOS or Open Drain output. |

Table 5. PSD9XX Pin Descriptions (cont.)

| Pin Name | Pin* (PLCC) | Type | Description |
|---|---|---|---|
| PC7 | 11 | I/O | PC7 pin of Port C. This port pin can be configured to have the following functions:<br>1. MCU I/O — write to or read from a standard output or input port.<br>2. Input to the PLDs.<br>3. DBE — active-low Data Byte Enable input from 68HC912 type MCUs.<br>This pin can be configured as a CMOS or Open Drain output. |
| PD0 | 10 | I/O | PD0 pin of Port D. This port pin can be configured to have the following functions:<br>1. ALE/AS input latches address output from the MCU.<br>2. MCU I/O — write or read from a standard output or input port.<br>3. Input to the PLDs.<br>4. General Purpose PLD output. |
| PD1 | 9 | I/O | PD1 pin of Port D. This port pin can be configured to have the following functions:<br>1. MCU I/O — write to or read from a standard output or input port.<br>2. Input to the PLDs.<br>3. General Purpose PLD output<br>4. CLKIN — clock input to the automatic power-down unit's power-down counter, and the PLD AND array. |
| PD2 | 8 | I/O | PD2 pin of Port D. This port pin can be configured to have the following functions:<br>1. MCU I/O — write to or read from a standard output or input port.<br>2. Input to the PLDs.<br>3. General Purpose PLD output.<br>4. $\overline{CSI}$ — chip select input. When low, the MCU can access the PSD memory and I/O. When high, the PSD memory blocks are disabled to conserve power. |
| $V_{CC}$ | 15, 38 | | Power pins |
| GND | 1,16,26 | | Ground pins |

*The pin numbers in this table are for the PLCC package only. See the package information section for pin numbers on other package types.

Table 6. I/O Port Latched Address Output Assignments*

| Microcontroller | Port A | | Port B | |
|---|---|---|---|---|
| | Port A (3:0) | Port A (7:4) | Port B (3:0) | Port B (7:4) |
| 8051XA (8-bit) | N/A | Address [7:4] | Address [11:8] | N/A |
| 80C251 (page mode) | N/A | N/A | Address [11:8] | Address [15:12] |
| All other 8-bit multiplexed | Address [3:0] | Address [7:4] | Address [3:0] | Address [7:4] |
| 8-bit non-multiplexed bus | N/A | N/A | Address [3:0] | Address [7:4] |

N/A = Not Applicable

*Refer to the I/O Port Section on how to enable the Latched Address Output function.

## 8.0 PSD9XX Register Description and Address Offset

Table 7 shows the offset addresses to the PSD9XX registers relative to the CSIOP base address. The CSIOP space is the 256 bytes of address that is allocated by the user to the internal PSD9XX registers. Table 7 provides brief descriptions of the registers in CSIOP space. For a more detailed description, refer to section 9.

Table 7. Register Address Offset

| Register Name | Port A | Port B | Port C | Port D | Other* | Description |
|---|---|---|---|---|---|---|
| Data In | 00 | 01 | 10 | 11 | | Reads Port pin as input, MCU I/O input mode |
| Control | 02 | 03 | | | | Selects mode between MCU I/O or Address Out |
| Data Out | 04 | 05 | 12 | 13 | | Stores data for output to Port pins, MCU I/O output mode |
| Direction | 06 | 07 | 14 | 15 | | Configures Port pin as input or output |
| Drive Select | 08 | 09 | 16 | 17 | | Configures Port pins as either CMOS or Open Drain on some pins, while selecting high slew rate on other pins. |
| Flash Protection | | | | | C0 | Read only – Flash Sector Protection |
| Secondary Flash Protection | | | | | C2 | Read only – PSD Security and Secondary Flash Sector Protection |
| PMMR0 | | | | | B0 | Power Management Register 0 |
| PMMR2 | | | | | B4 | Power Management Register 2 |
| Page | | | | | E0 | Page Register |
| VM | | | | | E2 | Places PSD memory areas in Program and/or Data space on an individual basis. |

*Other registers that are not part of the I/O ports.

# 9.0 The PSD9XX Functional Blocks

As shown in Figure 1, the PSD9XX consists of six major types of functional blocks:

- ❑ Memory Blocks
- ❑ PLD Blocks
- ❑ Bus Interface
- ❑ I/O Ports
- ❑ Power Management Unit
- ❑ JTAG Interface

The functions of each block are described in the following sections. Many of the blocks perform multiple functions, and are user configurable.

## 9.1 Memory Blocks

The PSD9XX has the following memory blocks:

- The main Flash memory
- Secondary Flash memory
- SRAM.

The memory select signals for these blocks originate from the Decode PLD (DPLD) and are user-defined in PSDsoft.

Table 8 summarizes which versions of the PSD9XX contain which memory blocks.

Table 8. Memory Blocks

| Device | Main Flash | | Secondary Flash Block | | SRAM |
|---|---|---|---|---|---|
| | Flash Size | Sector Size | Block Size | Sector Size | |
| PSD913F2 | 128KB | 16KB | 32KB | 8KB | 2KB |
| PSD934F2 | 256KB | 32KB | 32KB | 8KB | 8KB |
| PSD954F2 | 256KB | 32KB | 32KB | 8KB | 32KB |

### 9.1.1 Main Flash and Secondary Flash Memory Description

The main Flash memory block is divided evenly into eight sectors. The secondary Flash memory is divided into four sectors of eight Kbytes each. Each sector of either memory can be separately protected from program and erase operations.

Flash memory may be erased on a sector-by-sector basis and programmed byte-by-byte. Flash sector erasure may be suspended while data is read from other sectors of memory and then resumed after reading.

During a program or erase of Flash, the status can be output on the Rdy/Bsy pin of Port C3. This pin is set up using PSDsoft.

The PSD9XX Functional Blocks (cont.)

#### 9.1.1.1 Memory Block Selects

The decode PLD in the PSD9XX generates the chip selects for all the internal memory blocks (refer to the PLD section). Each of the eight Flash memory sectors have a Flash Select signal (FS0-FS7) which can contain up to three product terms. Each of the four secondary Flash memory sectors have a Select signal (CSBOOT0-3) which can contain up to three product terms. Having three product terms for each sector select signal allows a given sector to be mapped in different areas of system memory. When using a microcontroller with separate Program and Data space, these flexible select signals allow dynamic re-mapping of sectors from one space to the other when used with the VM Register (see section 9.1.3.1).

#### 9.1.1.2 The Ready/Busy Pin (PC3)

Pin PC3 can be used to output the Ready/Busy status of the PSD9XX. The output on the pin will be a '0' (Busy) when Flash memory blocks are being written to, **or** when the Flash memory block is being erased. The output will be a '1' (Ready) when no write or erase operation is in progress.

#### 9.1.1.3 Memory Operation

The main Flash and secondary Flash memories are addressed through the microcontroller interface on the PSD9XX device. The microcontroller can access these memories in one of two ways:

- The microcontroller can execute a typical bus write or read **operation** just as it would if accessing a RAM or ROM device using standard bus cycles.
- The microcontroller can execute a specific **instruction** that consists of several write and read operations. This involves writing specific data patterns to special addresses within the Flash to invoke an embedded algorithm. These instructions are summarized in Table 9.

Typically, Flash memory can be read by the microcontroller using read operations, just as it would read a ROM device. However, Flash memory can only be erased and programmed with specific instructions. For example, the microcontroller cannot write a single byte directly to Flash memory as one would write a byte to RAM. To program a byte into Flash memory, the microcontroller must execute a program instruction sequence, then test the status of the programming event. This status test is achieved by a read operation or polling the Rdy/Busy pin (PC3).

The Flash memory can also be read by using special instructions to retrieve particular Flash device information (sector protect status and ID).

The PSD9XX Functional Blocks (cont.)

9.1.1.3.1 Instructions

An instruction is defined as a sequence of specific operations. Each received byte is sequentially decoded by the PSD and not executed as a standard write operation. The instruction is executed when the correct number of bytes are properly received and the time between two consecutive bytes is shorter than the time-out value. Some instructions are structured to include read operations after the initial write operations.

The sequencing of any instruction must be followed exactly. Any invalid combination of instruction bytes or time-out between two consecutive bytes while addressing Flash memory will reset the device logic into a read array mode (Flash memory reads like a ROM device).

The PSD9XX main Flash and Secondary Flash support these instructions (see Table 9):

- ❏ Erase memory by chip or sector
- ❏ Suspend or resume sector erase
- ❏ Program a byte
- ❏ Reset to read array mode
- ❏ Read Main Flash Identifier value
- ❏ Read sector protection status
- ❏ Bypass Instruction (PSD934F2 and PSD954F2 only)

These instructions are detailed in Table 9. For efficient decoding of the instructions, the first two bytes of an instruction are the coded cycles and are followed by a command byte or confirmation byte. The coded cycles consist of writing the data AAh to address X555h during the first cycle and data 55h to address XAAAh during the second cycle. Address lines A15-A12 are don't care during the instruction write cycles. However, the appropriate sector select signal (FSi or CSBOOTi) must be selected.

The main Flash and the Secondary Flash Block have the same set of instructions (except Read main Flash ID). The chip selects of the Flash memory will determine which Flash will receive and execute the instruction. The main Flash is selected if any one of the FS0-7 is active, and the secondary Flash Block is selected if any one of the CSBOOT0-3 is active.

## The PSD9XX Functional Blocks (cont.)

Table 9. Instructions

| Instruction | FS0-7 or CSBOOT0-3 | Cycle 1 | Cycle 2 | Cycle 3 | Cycle 4 | Cycle5 | Cycle 6 | Cycle 7 |
|---|---|---|---|---|---|---|---|---|
| Read (Note 5) | 1 | "Read" RA RD | | | | | | |
| Read Main Flash ID (Notes 6,13) | 1 | AAh @555h | 55h @AAAh | 90h @555h | "Read" ID @x01h | | | |
| Read Sector Protection (Notes 6,8,13) | 1 | AAh @555h | 55h @AAAh | 90h @555h | "Read" 00h or 01h @x02h | | | |
| Program a Flash Byte | 1 | AAh @555h | 55h @AAAh | A0h @555h | PD@PA | | | |
| Erase One Flash Sector | 1 | AAh @555h | 55h @AAAh | 80h @555h | AAh @555h | 55h @AAAh | 30h @SA | 30h @next SA (Note 7) |
| Erase Flash Block (Bulk Erase) | 1 | AAh @555h | 55h @AAAh | 80h @555h | AAh @555h | 55h @AAAh | 10h @555h | |
| Suspend Sector Erase (Note 11) | 1 | B0h @xxxh | | | | | | |
| Resume Sector Erase (Note 12) | 1 | 30h @xxxh | | | | | | |
| Reset (Note 6) | 1 | F0 @ any address | | | | | | |
| Unlock Bypass (Note 14) | 1 | AAh @555h | 55h @AAAh | 20h @555h | | | | |
| Unlock Bypass Program (Note 9,14) | 1 | A0h @xxxh | PD@PA | | | | | |
| Unlock Bypass Reset (Note 10,14) | 1 | 90h @xxxh | 00h @xxxh | | | | | |

X = Don't Care.
RA = Address of the memory location to be read.
RD = Data read from location RA during read operation.
PA = Address of the memory location to be programmed. Addresses are latched on the falling edge of the WR# (CNTL0) pulse.
PD = Data to be programmed at location PA. Data is latched o the rising edge of WR# (CNTL0) pulse.
SA = Address of the sector to be erased or verified. The chip select (FS0-7 or CSBOOT0-3) of the sector to be erased must be active (high).

**NOTES:**
1. All bus cycles are write bus cycle except the ones with the "read" label.
2. All values are in hexadecimal.
3. FS0-7 and CSBOOT0-3 are active high and are defined in PSDsoft.
4. Only Address bits A11-A0 are used in Instruction decoding. A15-12 (or A16-A12) are don't care.
5. No unlock or command cycles required when device is in read mode.
6. The Reset command is required to return to the read mode after reading the Flash ID, Sector Protect status or if DQ5 (error flag) goes high.
7. Additional sectors to be erased must be entered within 80µs.
8. The data is 00h for an unprotected sector and 01h for a protected sector. In the fourth cycle, the sector chip select is active and (A1 = 1, A0 = 0).
9. The Unlock Bypass command is required prior to the Unlock Bypass Program command.
10. The Unlock Bypass Reset command is required to return to reading array data when the device is in the Unlock Bypass mode.
11. The system may read and program functions in non-erasing sectors, read the Flash ID or read the Sector Protect status, when in the Erase Suspend mode. The erase Suspend command is valid only during a sector erase operation.
12. The Erase Resume command is valid only during the Erase Suspend mode.
13. The MCU cannot invoke these instructions while executing code from the same Flash memory for which the instruction is intended. The MCU must fetch, for example, codes from the secondary block when reading the Sector Protection Status of the main Flash.
14. Available to PSD934F2 and PSD954F2 devices only.

The PSD9XX Functional Blocks (cont.)

#### 9.1.1.4 Power-Up Condition

The PSD9XX Flash memory is reset upon power-up to the read array mode. The FSi and CSBOOTi select signals, along with the write strobe signal, must be in the false state during power-up for maximum security of the data contents and to remove the possibility of a byte being written on the first edge of a write strobe signal. Any write cycle initiation is locked when $V_{CC}$ is below VLKO.

#### 9.1.1.5 Read

Under typical conditions, the microcontroller may read the Flash, or Secondary Flash memories using read operations just as it would a ROM or RAM device. Alternately, the microcontoller may use read operations to obtain status information about a program or erase operation in progress. Lastly, the microcontroller may use instructions to read special data from these memories. The following sections describe these read functions.

##### 9.1.1.5.1 Read the Contents of Memory

Main Flash and Secondary Flash memories are placed in the read array mode after power-up, chip reset, or a Reset Flash instruction (see Table 9). The microcontroller can read the memory contents of main Flash or Secondary Flash by using read operations any time the read operation is not part of an instruction sequence.

##### 9.1.1.5.2 Read the Main Flash Memory Identifier

The main Flash memory identifier is read with an instruction composed of 4 operations: 3 specific write operations and a read operation (see Table 9). During the read operation, address bits A6, A1, and A0 must be 0,0,1, respectively, and the appropriate sector select signal (FSi) must be active. The PSD9XX main Flash memory ID is E7h (PSD934/954F2) and E4h (PSD913F2).

##### 9.1.1.5.3 Read the Flash Memory Sector Protection Status

The Flash memory sector protection status is read with an instruction composed of 4 operations: 3 specific write operations and a read operation (see Table 9). During the read operation, address bits A6, A1, and A0 must be 0,1,0, respectively, while the chip select (FSi or CSBOOTi) designates the Flash sector whose protection has to be verified. The read operation will produce 01h if the Flash sector is protected, or 00h if the sector is not protected.

The sector protection status for all NVM blocks (main Flash or Secondary Flash) can also be read by the microcontroller accessing the Flash Protection and Secondary Flash Protection registers in PSD I/O space. See section 9.1.1.9.1 for register definitions.

##### 9.1.1.5.4 Read the Erase/Program Status Bits

The PSD9XX provides several status bits to be used by the microcontroller to confirm the completion of an erase or programming instruction of Flash memory. These status bits minimize the time that the microcontroller spends performing these tasks and are defined in Table 10. The status bits can be read as many times as needed.

Table 10. Status Bits

| | FSi/ CSBOOTi | DQ7 | DQ6 | DQ5 | DQ4 | DQ3 | DQ2 | DQ1 | DQ0 |
|---|---|---|---|---|---|---|---|---|---|
| Flash | $V_{IH}$ | Data Polling | Toggle Flag | Error Flag | X | Erase Time-out | X | X | X |

**NOTES:** 1. X = Not guaranteed value, can be read either 1 or 0.
2. DQ7-DQ0 represent the Data Bus bits, D7-D0.
3. FSi/CSBOOTi are active high.

For Flash memory, the microcontroller can perform a read operation to obtain these status bits while an erase or program instruction is being executed by the embedded algorithm. See section 9.1.1.7 for details.

The PSD9XX Functional Blocks (cont.)

#### 9.1.1.5.5 Data Polling Flag DQ7

When Erasing or Programming the Flash memory bit DQ7 outputs the complement of the bit being entered for Programming/Writing on DQ7. Once the Program instruction or the Write operation is completed, the true logic value is read on DQ7 (in a Read operation). Flash memory specific features:

- ❑ Data Polling is effective after the fourth Write pulse (for programming) or after the sixth Write pulse (for Erase). It must be performed at the address being programmed or at an address within the Flash sector being erased.
- ❑ During an Erase instruction, DQ7 outputs a '0'. After completion of the instruction, DQ7 will output the last bit programmed (it is a '1' after erasing).
- ❑ If the byte to be programmed is in a protected Flash sector, the instruction is ignored.
- ❑ If all the Flash sectors to be erased are protected, DQ7 will be set to '0' for about 100 µs, and then return to the previous addressed byte. No erasure will be performed.

#### 9.1.1.5.6 Toggle Flag DQ6

The PSD9XX offers another way for determining when the Flash memory Program instruction is completed. During the internal Write operation and when either the FSi or CSBOOTi is true, the DQ6 will toggle from '0' to '1' and '1' to '0' on subsequent attempts to read any byte of the memory.

When the internal cycle is complete, the toggling will stop and the data read on the Data Bus D0-7 is the addressed memory byte. The device is now accessible for a new Read or Write operation. The operation is finished when two successive reads yield the same output data. Flash memory specific features:

- ❑ The Toggle bit is effective after the fourth Write pulse (for programming) or after the sixth Write pulse (for Erase).
- ❑ If the byte to be programmed belongs to a protected Flash sector, the instruction is ignored.
- ❑ If all the Flash sectors selected for erasure are protected, DQ6 will toggle to '0' for about 100 µs and then return to the previous addressed byte.

#### 9.1.1.5.7 Error Flag DQ5

During a correct Program or Erase, the Error bit will set to '0'. This bit is set to '1' when there is a failure during Flash byte programming, Sector erase, or Bulk Erase.

In the case of Flash programming, the Error Bit indicates the attempt to program a Flash bit(s) from the programmed state (0) to the erased state (1), which is not a valid operation. The Error bit may also indicate a timeout condition while attempting to program a byte.

In case of an error in Flash sector erase or byte program, the Flash sector in which the error occurred or to which the programmed byte belongs must no longer be used. Other Flash sectors may still be used. The Error bit resets after the Reset instruction.

#### 9.1.1.5.8 Erase Time-out Flag DQ3

The Erase Timer bit reflects the time-out period allowed between two consecutive Sector Erase instructions. The Erase timer bit is set to '0' after a Sector Erase instruction for a time period of 100 µs + 20% unless an additional Sector Erase instruction is decoded. After this time period or when the additional Sector Erase instruction is decoded, DQ3 is set to '1'.

The PSD9XX Functional Blocks (cont.)

#### 9.1.1.6 Programming Flash Memory

Flash memory must be erased prior to being programmed. The MCU may erase Flash memory all at once or by-sector, but not byte-by-byte. A byte of Flash memory erases to all logic ones (FF hex), and its bits are programmed to logic zeros. Although erasing Flash memory occurs on a sector basis, programming Flash memory occurs on a byte basis.

The PSD9XX main Flash and Secondary Flash memories require the MCU to send an instruction to program a byte or perform an erase function (see Table 9).

Once the MCU issues a Flash memory program or erase instruction, it must check for the status of completion. The embedded algorithms that are invoked inside the PSD9XX support several means to provide status to the MCU. Status may be checked using any of three methods: Data Polling, Data Toggle, or the Ready/Busy output pin.

##### 9.1.1.6.1 Data Polling

Polling on DQ7 is a method of checking whether a Program or Erase instruction is in progress or has completed. Figure 3 shows the Data Polling algorithm.

When the MCU issues a programming instruction, the embedded algorithm within the PSD9XX begins. The MCU then reads the location of the byte to be programmed in Flash to check status. Data bit DQ7 of this location becomes the compliment of data bit 7of the original data byte to be programmed. The MCU continues to poll this location, comparing DQ7 and monitoring the Error bit on DQ5. When the DQ7 matches data bit 7 of the original data, and the Error bit at DQ5 remains ‘0’, then the embedded algorithm is complete.
If the Error bit at DQ5 is ‘1’, the MCU should test DQ7 again since DQ7 may have changed simultaneously with DQ5 (see Figure 3).

The Error bit at DQ5 will be set if either an internal timeout occurred while the embedded algorithm attempted to program the byte or if the MCU attempted to program a ‘1’ to a bit that was not erased (not erased is logic ‘0’).

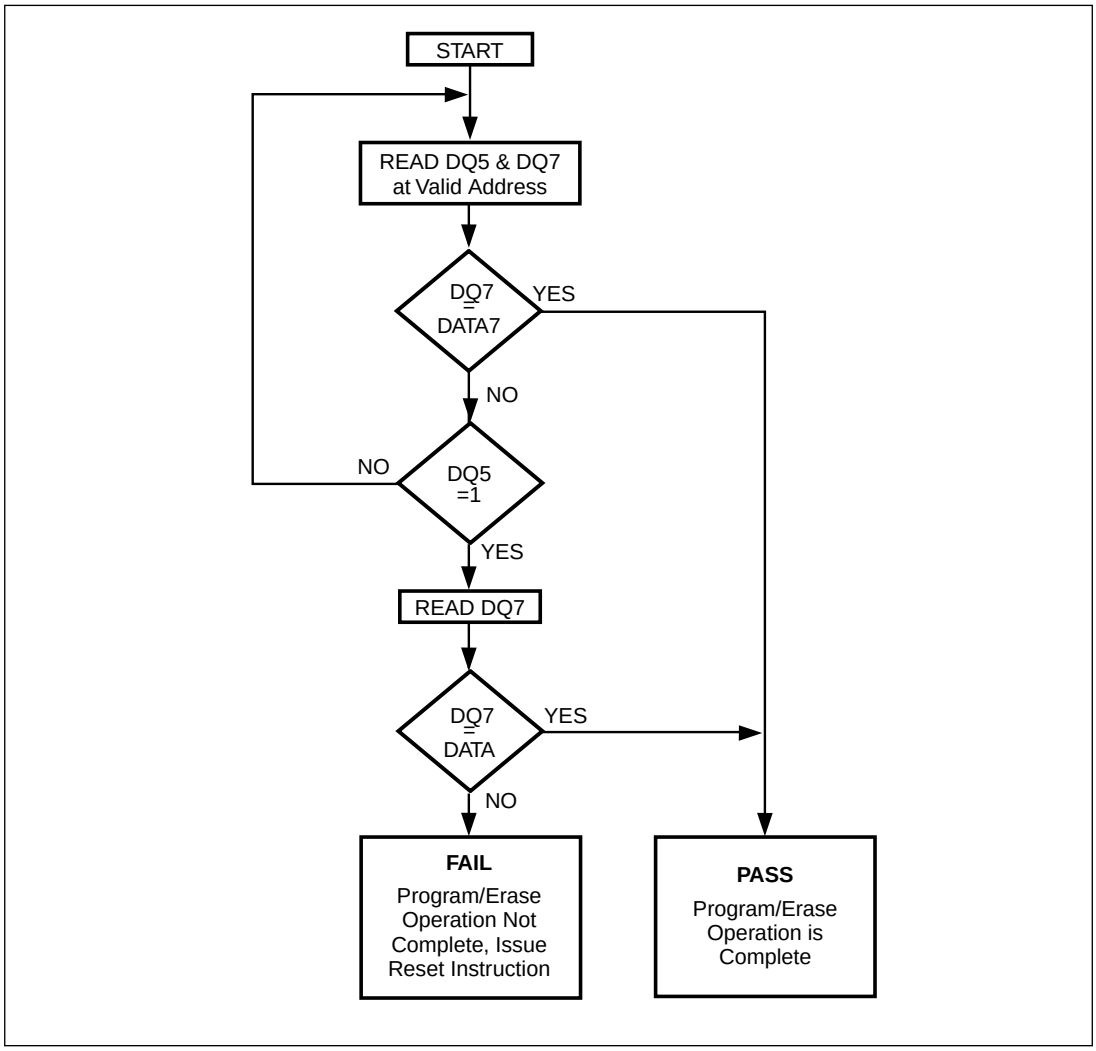
It is suggested (as with all Flash memories) to read the location again after the embedded programming algorithm has completed to compare the byte that was written to Flash with the byte that was intended to be written.

When using the Data Polling method after an erase instruction, Figure 3 still applies. However, DQ7 will be ‘0’ until the erase operation is complete. A ‘1’ on DQ5 will indicate a timeout failure of the erase operation, a ‘0’ indicates no error. The MCU can read any location within the sector being erased to get DQ7 and DQ5.

PSDsoft will generate ANSI C code functions which implement these Data Polling algorithms.

The PSD9XX Functional Blocks (cont.)

Figure 3. Data Polling Flow Chart




9.1.1.6.2 Data Toggle

Checking the Data Toggle bit on DQ6 is a method of determining whether a Program or Erase instruction is in progress or has completed. Figure 4 shows the Data Toggle algorithm.

When the MCU issues a programming instruction, the embedded algorithm within the PSD9XX begins. The MCU then reads the location of the byte to be programmed in Flash to check status. Data bit DQ6 of this location will toggle each time the MCU reads this location until the embedded algorithm is complete. The MCU continues to read this location, checking DQ6 and monitoring the Error bit on DQ5. When DQ6 stops toggling (two consecutive reads yield the same value), and the Error bit on DQ5 remains '0', then the embedded algorithm is complete. If the Error bit on DQ5 is '1', the MCU should test DQ6 again, since DQ6 may have changed simultaneously with DQ5 (see Figure 4).

The Error bit at DQ5 will be set if either an internal timeout occurred while the embedded algorithm attempted to program the byte, or if the MCU attempted to program a '1' to a bit that was not erased (not erased is logic '0').

The PSD9XX Functional Blocks (cont.)

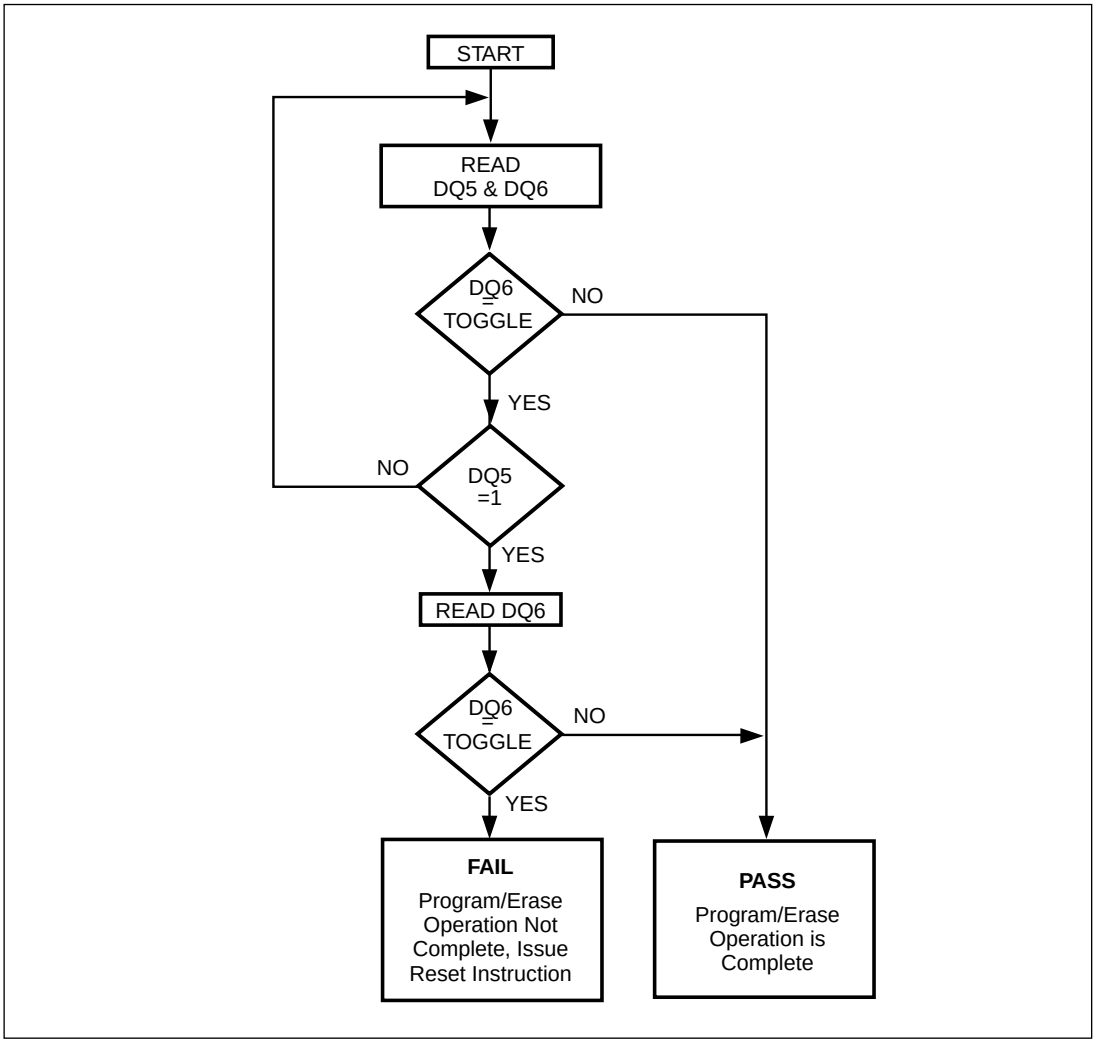
9.1.1.6.2 Data Toggle (cont.)

It is suggested (as with all Flash memories) to read the location again after the embedded programming algorithm has completed to compare the byte that was written to Flash with the byte that was intended to be written.

When using the Data Toggle method after an erase instructin, Figure 4 still applies. DQ6 will toggle until the erase operation is complete. A '1' on DQ5 will indicate a timeout failure of the erase operation, a '0' indicates no error. The MCU can read any location within the sector being erased to get DQ6 and DQ5.

PSDsoft will generate ANSI C code functions which implement these Data Toggling algorithms.

Figure 4. Data Toggle Flow Chart



The PSD9XX Functional Blocks (cont.)

#### 9.1.1.7 Unlock Bypass Instruction (PSD934F2 and PSD954F2 only)

The unlock bypass feature allows the system to program bytes to the flash memories faster than using the standard program instruction. The unlock bypass instruction is initiated by first writing two unlock cycles. This is followed by a third write cycle containing the unlock bypass command, 20h (see Table 9). The flash memory then enters the unlock bypass mode. A two-cycle Unlock Bypass Program instruction is all that is required to program in this mode. The first cycle in this instruction contains the unlock bypass programm command, A0h; the second cycle contains the program address and data. Additional data is programmed in the same manner. This mode dispenses with the initial two unlock cycles requiredc in the standard program instruction, resulting in faster total programming time. During the unlock bypass mode, only the Unlock Bypass Program and Unlock Bypass Reset instructions are valid. To exit the unlock bypass mode, the system must issue the two-cycle unlock bypass reset instruction. The first cycle must contain the data 90h; the second cycle the data 00h. Addresses are don't care for both cycles. The falsh memory then returns to reading array data mode.

#### 9.1.1.8 Erasing Flash Memory

##### 9.1.1.8.1. Flash Bulk Erase Instruction

The Flash Bulk Erase instruction uses six write operations followed by a Read operation of the status register, as described in Table 9. If any byte of the Bulk Erase instruction is wrong, the Bulk Erase instruction aborts and the device is reset to the Read Flash memory status.

During a Bulk Erase, the memory status may be checked by reading status bits DQ5, DQ6, and DQ7, as detailed in section 9.1.1.6. The Error bit (DQ5) returns a '1' if there has been an Erase Failure (maximum number of erase cycles have been executed).

It is not necessary to program the array with 00h because the PSD9XX will automatically do this before erasing to 0FFh.

During execution of the Bulk Erase instruction, the Flash memory will not accept any instructions.

##### 9.1.1.8.2 Flash Sector Erase Instruction

The Sector Erase instruction uses six write operations, as described in Table 9. Additional Flash Sector Erase confirm commands and Flash sector addresses can be written subsequently to erase other Flash sectors in parallel, without further coded cycles, if the additional instruction is transmitted in a shorter time than the timeout period of about 100 µs. The input of a new Sector Erase instruction will restart the time-out period.

The status of the internal timer can be monitored through the level of DQ3 (Erase time-out bit). If DQ3 is '0', the Sector Erase instruction has been received and the timeout is counting. If DQ3 is '1', the timeout has expired and the PSD9XX is busy erasing the Flash sector(s). Before and during Erase timeout, any instruction other than Erase suspend and Erase Resume will abort the instruction and reset the device to Read Array mode.
It is not necessary to program the Flash sector with 00h as the PSD9XX will do this automatically before erasing (byte=FFh).

During a Sector Erase, the memory status may be checked by reading status bits DQ5, DQ6, and DQ7, as detailed in section 9.1.1.6.

During execution of the erase instruction, the Flash block logic accepts only Reset and Erase Suspend instructions. Erasure of one Flash sector may be suspended, in order to read data from another Flash sector, and then resumed.

The PSD9XX Functional Blocks (cont.)

#### 9.1.1.8.3 Flash Erase Suspend Instruction

When a Flash Sector Erase operation is in progress, the Erase Suspend instruction will suspend the operation by writing 0B0h to any address when an appropriate Chip Select (FSi or CSBOOTi) is true. (See Table 9). This allows reading of data from another Flash sector after the Erase operation has been suspended. Erase suspend is accepted only during the Flash Sector Erase instruction execution and defaults to read array mode. An Erase Suspend instruction executed during an Erase timeout will, in addition to suspending the erase, terminate the time out.

The Toggle Bit DQ6 stops toggling when the PSD9XX internal logic is suspended. The toggle Bit status must be monitored at an address within the Flash sector being erased. The Toggle Bit will stop toggling between 0.1 µs and 15 µs after the Erase Suspend instruction has been executed. The PSD9XX will then automatically be set to Read Flash Block Memory Array mode.

If an Erase Suspend instruction was executed, the following rules apply:

- Attempting to read from a Flash sector that was being erased will output invalid data.
- Reading from a Flash sector that was **not** being erased is valid.
- The Flash memory **cannot** be programmed, and will only respond to Erase Resume and Reset instructions (read is an operation and is OK).
- If a Reset instruction is received, data in the Flash sector that was being erased will be invalid.

#### 9.1.1.8.4 Flash Erase Resume Instruction

If an Erase Suspend instruction was previously executed, the erase operation may be resumed by this instruction. The Erase Resume instruction consists of writing 030h to any address while an appropriate Chip Select (FSi or CSBOOTi) is true. (See Table 9.)

### 9.1.1.9 Specific Features

#### 9.1.1.9.1 Flash and Secondary Flash Sector Protect

Each Flash and Secondary Flash sector can be separately protected against Program and Erase functions. Sector Protection provides additional data security because it disables all program or erase operations. This mode can be activated through the JTAG Port or a Device Programmer.

Sector protection can be selected for each sector using the PSDsoft Configuration program. This will automatically protect selected sectors when the device is programmed through the JTAG Port or a Device Programmer. Flash sectors can be unprotected to allow updating of their contents using the JTAG Port or a Device Programmer. The microcontroller can read (but cannot change) the sector protection bits.

Any attempt to program or erase a protected Flash sector will be ignored by the device. The Verify operation will result in a read of the protected data. This allows a guarantee of the retention of the Protection status.

The sector protection status can be read by the MCU through the Flash protection and Secondary Flash protection registers (CSIOP). See Table 11.

The PSD9XX Functional Blocks (cont.)

Table 11. Sector Protection/Security Bit Definition

Flash Protection Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| Sec7_Prot | Sec6_Prot | Sec5_Prot | Sec4_Prot | Sec3_Prot | Sec2_Prot | Sec1_Prot | Sec0_Prot |

Bit Definitions:

**Sec<i>_Prot** 1 = Main Flash Sector <i> is write protected.
**Sec<i>_Prot** 0 = Main Flash Sector <i> is not write protected.

Secondary Flash Protection Register

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| Security_ Bit | * | * | * | Sec3_Prot | Sec2_Prot | Sec1_Prot | Sec0_Prot |

*: Not used.

Bit Definitions:

**Sec<i>_Prot** 1 = Secondary Flash Sector <i> is write protected.
**Sec<i>_Prot** 0 = Secondary Flash Sector <i> is not write protected.

**Security_Bit** 0 = Security Bit in device has not been set.
1 = Security Bit in device has been set.

#### 9.1.1.9.2 Reset Instruction – PSD913F2

The Reset instruction consists of one write cycle (see Table 9). It can also be optionally preceded by the standard two write decoding cycles (writing AAh to 555h and 55h to AAAh).

The Reset instruction must be executed after:

1. Reading the Flash Protection status or Flash ID
2. When an error condition occurs (DQ5 goes high) during a Flash programming or erase cycle.

The Reset instruction will reset the Flash to normal Read Mode. It may take the Flash memory up to few msec to complete the reset cycle.

The Reset instruction is ignored when it is issued during a Flash programming or Bulk Erase cycle. During Sector Erase cycle, the Reset instruction will abort the on going sector erase cycle and return the Flash to normal Read Mode in up to few msec.

#### 9.1.1.9.3 Reset Instruction – PSD934F2, PSD954F2

The Reset instruction consists of one write cycle (see Table 9). It can also be optionally preceded by the standard two write decoding cycles (writing AAh to 555h and 55h to AAAh).

The Reset instruction must be executed after:

1. Reading the Flash Protection status or Flash ID
2. When an error condition occurs (DQ5 goes high) during a Flash programming or erase cycle.

The Reset instruction will immediately reset the Flash to normal Read Mode. However, if there is an error condition (DQ5 goes high), the Flash memory will return to the Read Mode in 25 µsec after the Reset instruction is issued.

The Reset instruction is ignored when it is issued during a Flash programming or Bulk Erase cycle. The Reset instruction will abort the on going sector erase cycle and return the Flash memory to normal Read Mode in 25 µsec.

The PSD9XX Functional Blocks (cont.)

9.1.1.9.4 Reset Pin Input – PSD934F2, PSD954F2

The reset pulse input from the pin will abort any operation in progress and reset the Flash memory to Read Mode. When the reset occurs during a programming or erase cycle, the Flash memory will take up to 25 µsec to return to Read Mode. It is recommended that the reset pulse (except power on reset, see Reset Section) be at least 25 µSec such that the Flash memory will always be ready for the MCU to fetch the boot codes after reset is over.

### 9.1.2 SRAM

The SRAM is enabled when RS0—the SRAM chip select output from the DPLD—is high. RS0 can contain up to two product terms, allowing flexible memory mapping.

The SRAM can be backed up using an external battery. The external battery should be connected to the Vstby pin (PC2). If you have an external battery connected to the PSD9XX, the contents of the SRAM will be retained in the event of a power loss. The contents of the SRAM will be retained so long as the battery voltage remains at 2V or greater. If the supply voltage falls below the battery voltage, an internal power switchover to the battery occurs.

Pin PC4 can be configured as an output that indicates when power is being drawn from the external battery. This Vbaton signal will be high with the supply voltage falls below the battery voltage and the battery on PC2 is supplying power to the internal SRAM.

The chip select signal (RS0) for the SRAM, Vstby, and Vbaton are all configured using PSDsoft.

### 9.1.3 Memory Select Signals

The main Flash (FSi), Secondary Flash (CSBOOTi), and SRAM (RS0) memory select signals are all outputs of the DPLD. They are setup by entering equations for them in PSDsoft. The following rules apply to the equations for the internal chip select signals:

1. Main Flash memory and Secondary Flash memory sector select signals must **not** be larger than the physical sector size.
2. Any main Flash memory sector must **not** be mapped in the same memory space as another Main Flash sector.
3. A Secondary Flash memory sector must **not** be mapped in the same memory space as another Secondary Flash sector.
4. SRAM, I/O, and Peripheral I/O spaces must **not** overlap.
5. A Secondary Flash memory sector **may** overlap a main Flash memory sector. In case of overlap, priority will be given to the Secondary Flash sector.
6. SRAM, I/O, and Peripheral I/O spaces **may** overlap any other memory sector. Priority will be given to the SRAM, I/O, or Peripheral I/O.

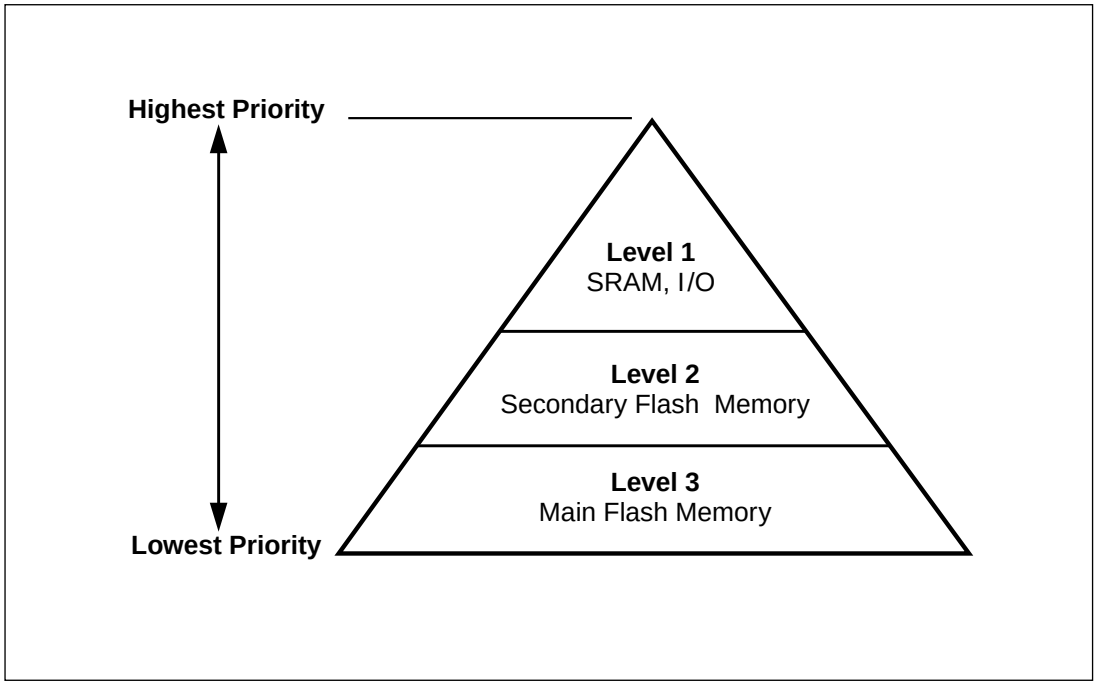
Example

FS0 is valid when the address is in the range of 8000h to BFFFh, CSBOOT0 is valid from 8000h to 9FFFh, and RS0 is valid from 8000h to 87FFh. Any address in the range of RS0 will always access the SRAM. Any address in the range of CSBOOT0 greater than 87FFh (and less than 9FFFh) will automatically address Boot memory segment 0. Any address greater than 9FFFh will access the Flash memory segment 0. You can see that half of the Flash memory segment 0 and one-fourth of Boot segment 0 can not be accessed in this example. Also note that an equation that defined FS1 to anywhere in the range of 8000h to BFFFh would **not** be valid.

Figure 5 shows the priority levels for all memory components. Any component on a higher level can overlap and has priority over any component on a lower level. Components on the same level must **not** overlap. Level one has the highest priority and level 3 has the lowest.

The PSD9XX Functional Blocks (cont.)

Figure 5. Priority Level of Memory and I/O Components




9.1.3.1. Memory Select Configuration for MCUs with Separate Program and Data Spaces

The 8031 and compatible family of microcontrollers, which includes the 80C51, 80C151, 80C251, 80C51XA, and the C500 family have separate address spaces for code memory (selected using PSEN) and data memory (selected using RD). Any of the memories within the PSD9XX can reside in either space or both spaces. This is controlled through manipulation of the VM register that resides in the PSD's CSIOP space.

The VM register is set using PSDsoft to have an initial value. It can subsequently be changed by the microcontroller so that memory mapping can be changed on-the-fly. For example, you may wish to have SRAM and Flash in Data Space at boot, and Boot Block in Program Space at boot, and later swap Boot Block and Flash. This is easily done with the VM register by using PSDsoft to configure it for boot up and having the microcontroller change it when desired.

Table 13 describes the VM Register.

Table 13. VM Register

| Bit 7* | Bit 6* | Bit 5* | Bit 4 FL_Data | Bit 3 Boot_Data | Bit 2 FL_Code | Bit 1 Boot_Code | Bit 0 SRAM_Code |
|---|---|---|---|---|---|---|---|
| * | * | * | 0 = RD can't access Flash | 0 = RD can't access Secondary Flash | 0 = PSEN can't access Flash | 0 = PSEN can't access Secondary Flash | 0 = PSEN can't access SRAM |
| * | * | * | 1 = RD access Flash | 1 = RD access Secondary Flash | 1 = PSEN access Flash | 1 = PSEN access Secondary Flash | 1 = PSEN access SRAM |

**NOTE:** Bits 5-7 are not used, should set to "0".

The PSD9XX Functional Blocks (cont.)

9.1.3.2 Configuration Modes for MCUs with Separate Program and Data Spaces

9.1.3.2.1 Separate Space Modes

Code memory space is separated from data memory space. For example, the PSEN signal is used to access the program code from the Flash Memory, while the RD signal is used to access data from the Boot memory, SRAM and I/O Ports. This configuration requires the VM register to be set to 0Ch.

9.1.3.2.2 . Combined Space Modes

The program and data memory spaces are combined into one space that allows the main Flash Memory, Boot memory, and SRAM to be accessed by either PSEN or RD. For example, to configure the main Flash memory in combined space mode, bits 2 and 4 of the VM register are set to “1”.

9.1.3.3 80C31 Memory Map Example

See Application Note for examples.

Figure 6. 8031 Memory Modes – Separate Space Mode

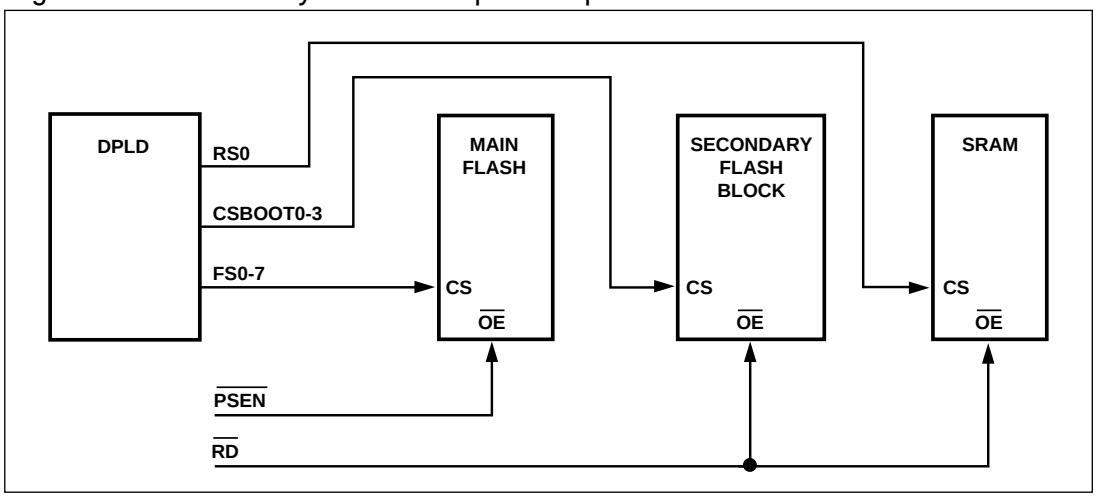
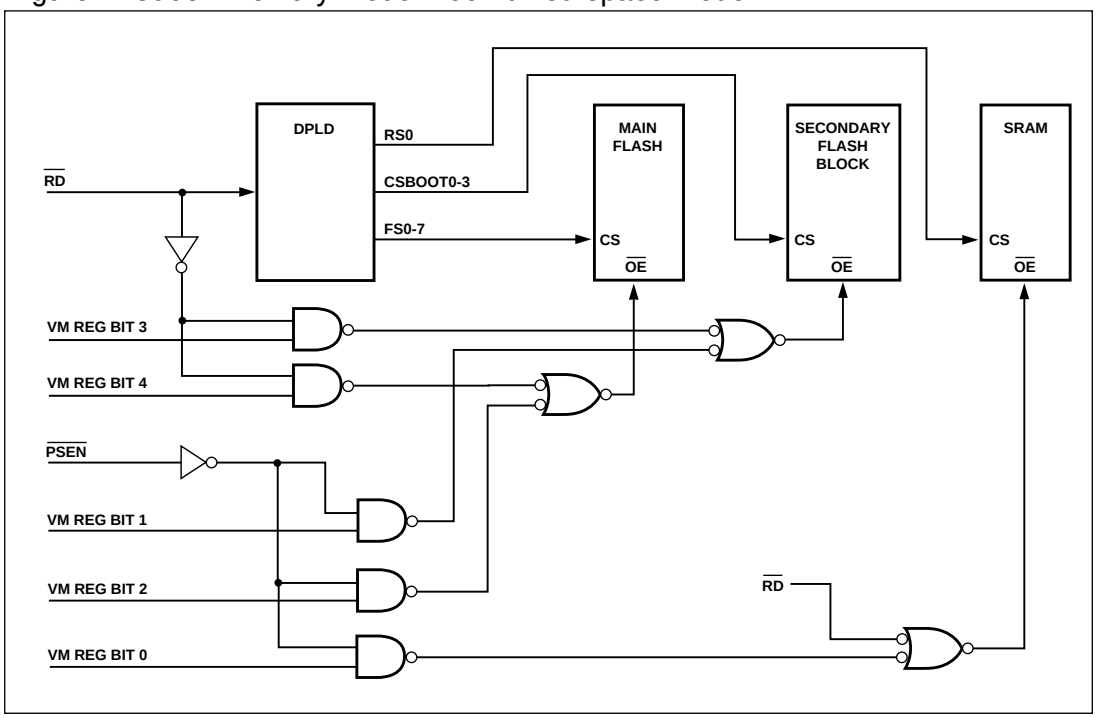



Figure 7. 80C31 Memory Mode – Combined Space Mode



The PSD9XX Functional Blocks (cont.)

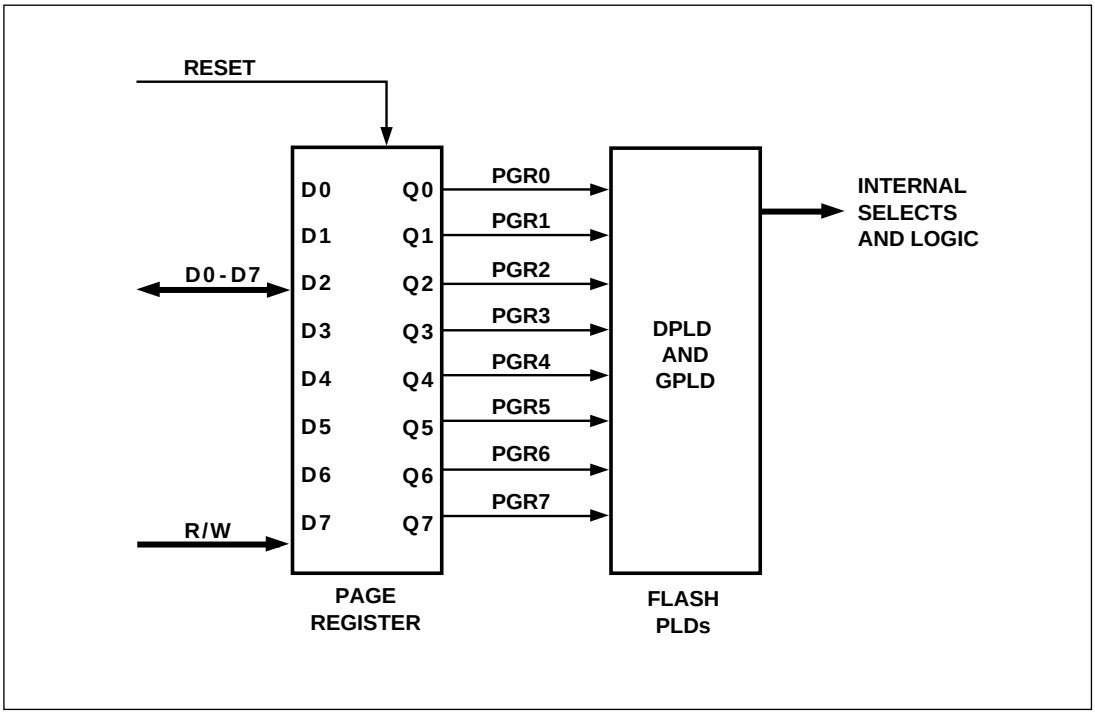
### 9.1.4 Page Register

The eight bit Page Register increases the addressing capability of the microcontroller by a factor of up to 256. The contents of the register can also be read by the microcontroller. The outputs of the Page Register (PGR0-PGR7) are inputs to the PLD and can be included in the Flash Memory, Secondary Flash Block, and SRAM chip select equations.

If memory paging is not needed, or if not all 8 page register bits are needed for memory paging, then these bits may be used in the PLD for general logic. See Application Note.

Figure 8 shows the Page Register. The eight flip flops in the register are connected to the internal data bus D0-D7. The microcontroller can write to or read from the Page Register. The Page Register can be accessed at address location CSIOP + E0h.

Figure 8. Page Register



The PSD9XX Functional Blocks (cont.)

## 9.2 PLDs

The PLDs bring programmable logic functionality to the PSD9XX. After specifying the chip selects or logic equations for the PLDs in PSDsoft, the logic is programmed into the device and available upon power-up.

The PSD9XX contains two PLDs: the Decode PLD (DPLD), and the General Purpose PLD (GPLD). The PLDs are briefly discussed in the next few paragraphs, and in more detail in sections 9.2.1 and 9.2.2. Figure 10 shows the configuration of the PLDs.

The DPLD performs address decoding for internal components, such as memory, registers, and I/O port selects.

The GPLD can be used to generate external chip selects, control signals or logic functions. The GPLD has 19 outputs that are connected to Ports A, B and D.

The AND array is used to form product terms. These product terms are specified using PSsoft. An Input Bus consisting of 57 signals is connected to the PLDs. The signals are shown in Table 15. The complement of the 57 signals are also available as input to the AND array.

Table 15. DPLD and GPLD Inputs

| Input Source | Input Name | Number of Signals |
|---|---|---|
| MCU Address Bus | A[15:0]* | 16 |
| MCU Control Signals | CNTL[2:0] | 3 |
| Reset | RST | 1 |
| Power Down | PDN | 1 |
| Port A Input | PA[7-0] | 8 |
| Port B Input | PB[7-0] | 8 |
| Port C Input | PC[7-0] | 8 |
| Port D Inputs | PD[2:0] | 3 |
| Page Register | PGR(7:0) | 8 |
| Flash Programming Status Bit | Rdy/Bsy | 1 |

**NOTE:** The address inputs are A[19:4] in 80C51XA mode.

### The Turbo Bit

The PLDs in the PSD9XX can minimize power consumption by switching off when inputs remain unchanged for an extended time of about 70 ns. Setting the Turbo mode bit to off (Bit 3 of the PMMR0 register) automatically places the PLDs into standby if no inputs are changing. Turbo-off mode increases propagation delays while reducing power consumption. Refer to the Power Management Unit section on how to set the Turbo Bit. Additionally, five bits are available in the PMMR2 register to block MCU control signals from entering the PLDs. This reduces power consumption and can be used only when these MCU control signals are not used in PLD logic equations.

Figure 9. PLD Block Diagrams

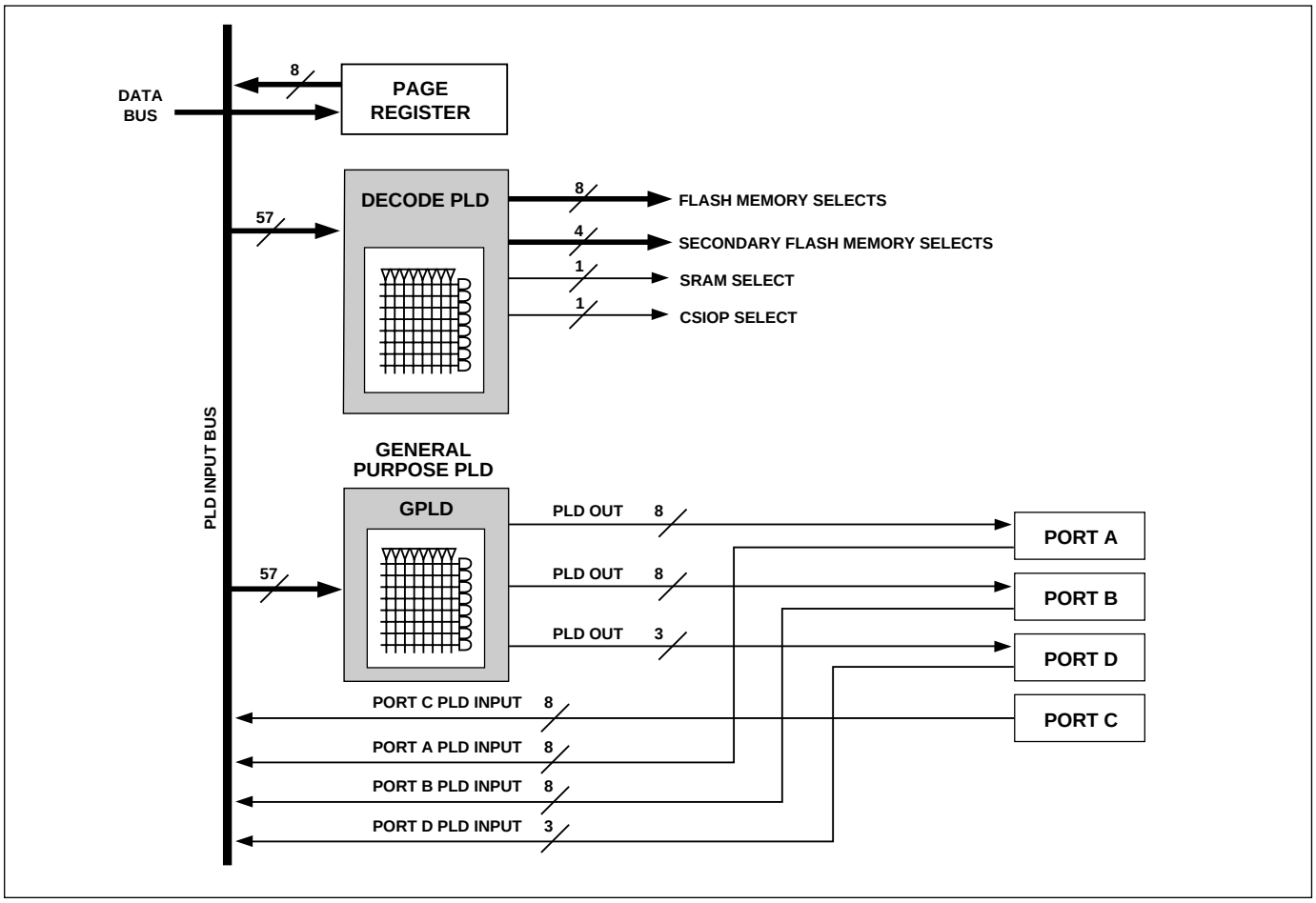
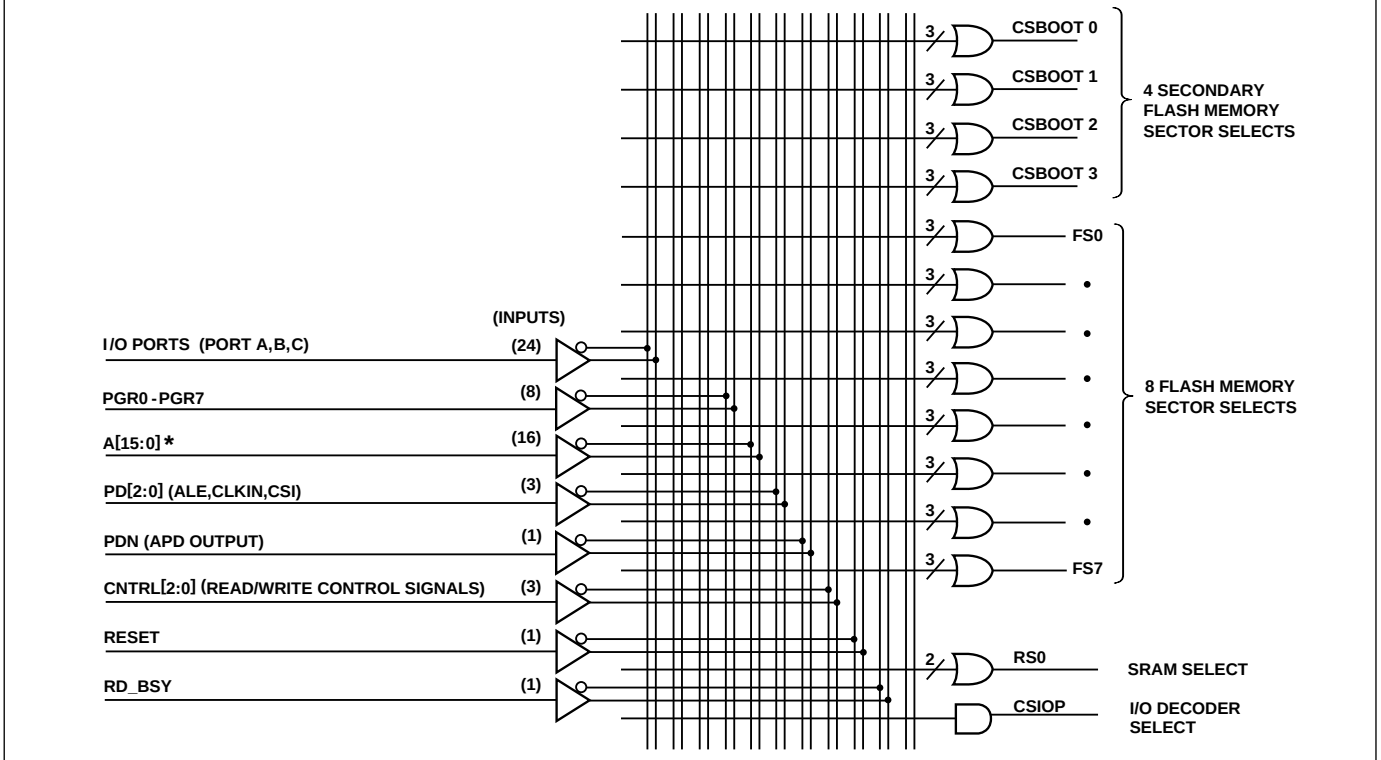



Figure 10. DPLD Logic Array




*NOTE: The address inputs are A[19:4] in 80C51XA mode.

The PSD9XX Functional Blocks (cont.)

### 9.2.1 Decode PLD (DPLD)

The DPLD, shown in Figure 10, is used for decoding the address for internal PSD components. The DPLD can generate the following chip selects:

- 8 sector selects for the main Flash memory (three product terms each)
- 4 sector selects for the Secondary Flash memory (three product terms each)
- 1 internal SRAM select (two product terms)
- 1 internal CSIOP select (select PSD registers, one product term)

Inputs to the DPLD chip selects may include address inputs, Page Register inputs and other user defined external inputs from Ports A, B, C or D.

### 9.2.2 General Purpose PLD (GPLD)

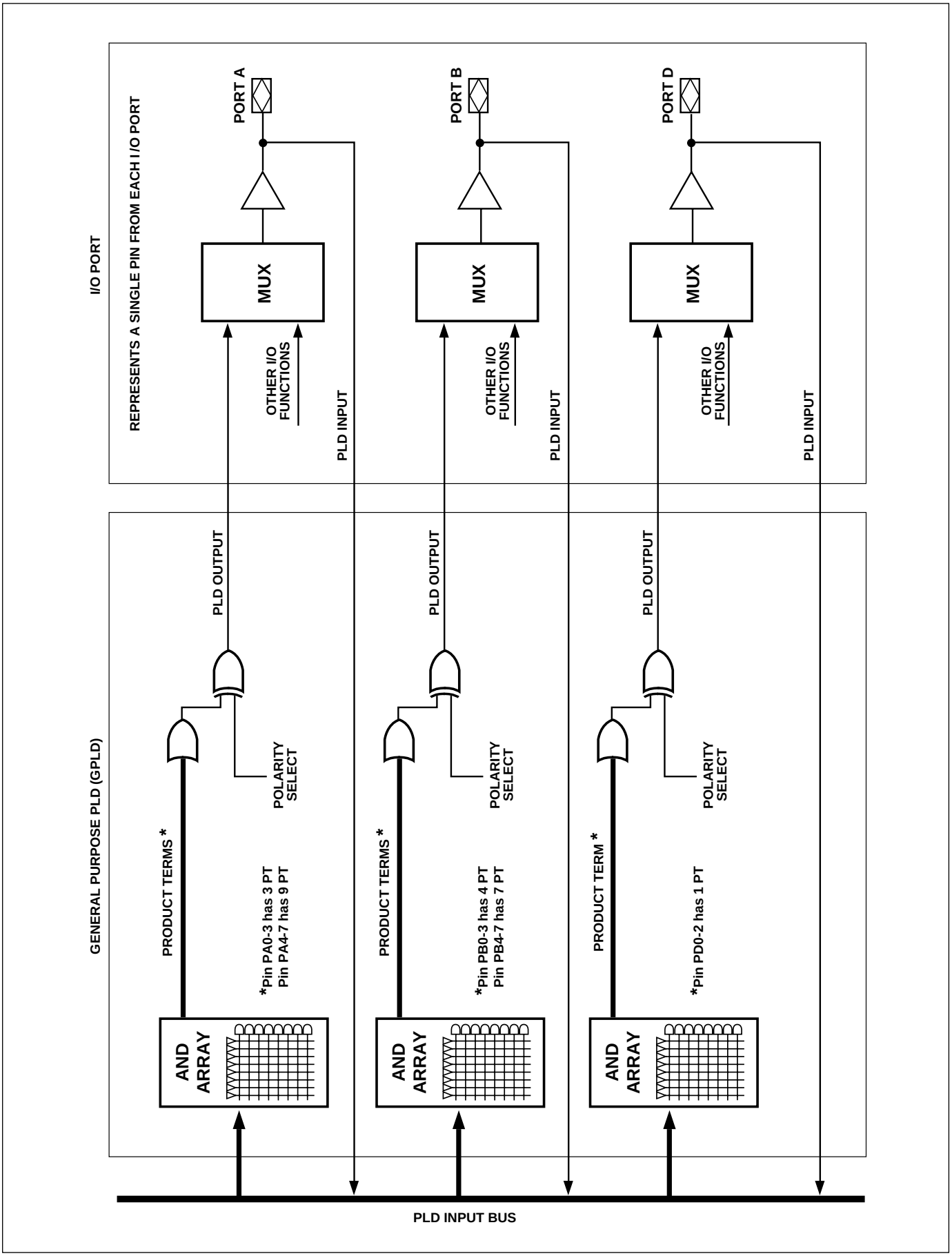
The General Purpose PLD implements user defined system combinatorial logic function or chip selects for external devices. Figure 11 shows how the GPLD is connected to the I/O Ports. The GPLD has 19 outputs and each are routed to a port pin. The port pin can also be configured as input tot eh GPLD. When it is not used as GPLD output or input, the pin can be configured to perform other I/O functions.

The GPLD outputs are identical except in the number of available product terms for logic implementation. Select the pin that can best meet the product term requirement of your logic function or chip selects. The outputs can be configured as active high or low outputs. Table 16 shows the number of product terms that are assigned to the PLD outputs on the I/O Ports. When PSD9XX is connected to a MCU with non-multiplexed bus, Port A will be configured as the Data Port and the GPLD outputs will not be available.

Table 16. GPLD Output Product Term

| GPLD Output on Port Pin | Number of Product Terms |
|---|---|
| Port A, pins PA0-3 | 3 |
| Port A, pins PA4-7 | 9 |
| Port B, pins PB0-3 | 4 |
| Port B, pins PB4-7 | 7 |
| Port D, pins PD0-2 | 1 |

Figure 11. General Purpose PLD and I/O Port



The PSD9XX Functional Blocks (cont.)

## 9.3 Microcontroller Bus Interface

The “no-glue logic” PSD9XX Microcontroller Bus Interface can be directly connected to most popular microcontrollers and their control signals. Key 8-bit microcontrollers with their bus types and control signals are shown in Table 17. The interface type is specified using the PSDsoft.

Table 17. Microcontrollers and their Control Signals

| MCU | Data Bus Width | CNTL0 | CNTL1 | CNTL2 | PC7 | PD0** | ADIO0 | PA3-PA0 | PA7-PA4 |
|---|---|---|---|---|---|---|---|---|---|
| 8031/8051 | 8 | $\overline{WR}$ | $\overline{RD}$ | $\overline{PSEN}$ | * | ALE | A0 | * | * |
| 80C51XA | 8 | $\overline{WR}$ | $\overline{RD}$ | $\overline{PSEN}$ | * | ALE | A4 | A3-A0 | * |
| 80C251 | 8 | $\overline{WR}$ | $\overline{PSEN}$ | * | * | ALE | A0 | * | * |
| 80C251 | 8 | $\overline{WR}$ | $\overline{RD}$ | $\overline{PSEN}$ | * | ALE | A0 | * | * |
| 80198 | 8 | $\overline{WR}$ | $\overline{RD}$ | * | * | ALE | A0 | * | * |
| 68HC11 | 8 | R/$\overline{W}$ | E | * | * | AS | A0 | * | * |
| 68HC05C0 | 8 | $\overline{WR}$ | $\overline{RD}$ | * | * | AS | A0 | * | * |
| 68HC912 | 8 | $\overline{R/W}$ | E | * | $\overline{DBE}$ | AS | A0 | * | * |
| Z80 | 8 | WR | $\overline{RD}$ | * | * | * | A0 | D3-D0 | D7-D4 |
| Z8 | 8 | R/$\overline{W}$ | $\overline{DS}$ | * | * | $\overline{AS}$ | A0 | * | * |
| 68330 | 8 | R/$\overline{W}$ | $\overline{DS}$ | * | * | AS | A0 | * | * |
| M37702M2 | 8 | R/$\overline{W}$ | $\overline{E}$ | * | * | ALE | A0 | D3-D0 | D7-D4 |

*Unused CNTL2 pin can be configured as PLD input. Other unused pins (PC7, PD0, PA3-0) can be configured for other I/O functions.

**ALE/AS input is optional for microcontrollers with a non-multiplexed bus

### 9.3.1. PSD9XX Interface to a Multiplexed 8-Bit Bus

Figure 12 shows an example of a system using a microcontroller with an 8-bit multiplexed bus and a PSD9XX. The ADIO port on the PSD9XX is connected directly to the microcontroller address/data bus. ALE latches the address lines internally. Latched addresses can be brought out to Port A or B. The PSD9XX drives the ADIO data bus only when one of its internal resources is accessed and the RD input is active. Should the system address bus exceed sixteen bits, Ports A, B, C, or D may be used as additional address inputs.

### 9.3.2. PSD9XX Interface to a Non-Multiplexed 8-Bit Bus

Figure 13 shows an example of a system using a microcontroller with an 8-bit non-multiplexed bus and a PSD9XX. The address bus is connected to the ADIO Port, and the data bus is connected to Port A. Port A is in tri-state mode when the PSD9XX is not accessed by the microcontroller. Should the system address bus exceed sixteen bits, Ports B, C, or D may be used for additional address inputs.

## The PSD9XX Functional Blocks (cont.)

Figure 12. An Example of a Typical 8-Bit Multiplexed Bus Interface

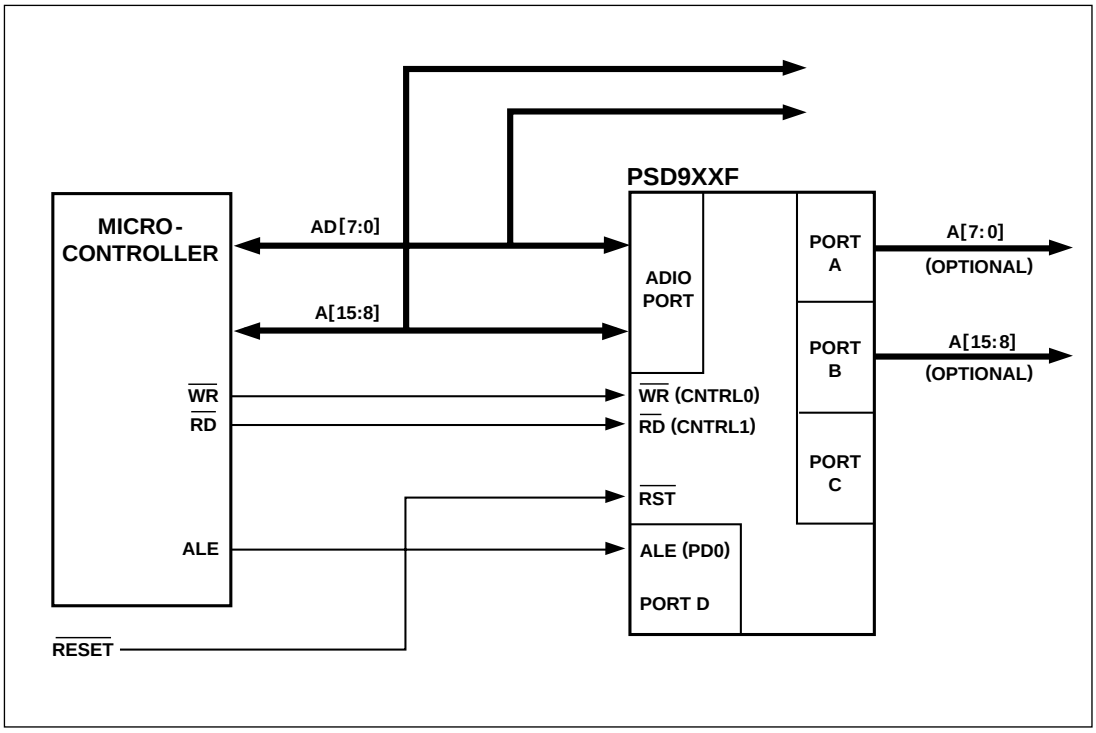
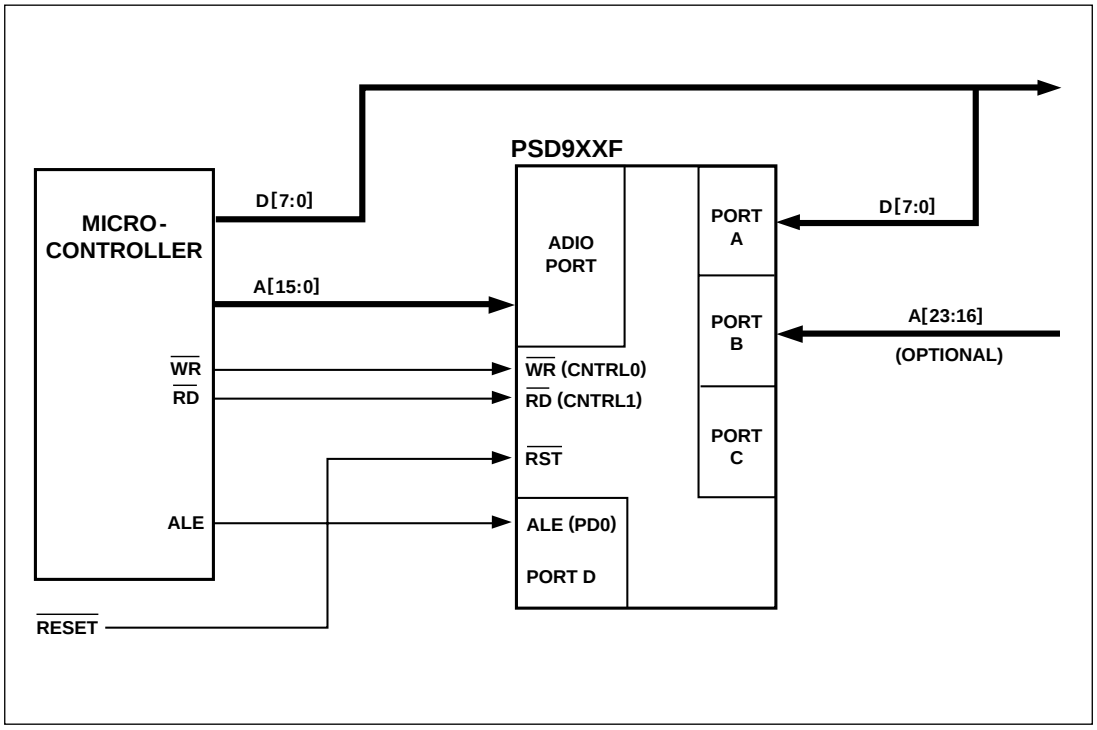



Figure 13. An Example of a Typical 8-Bit Non-Multiplexed Bus Interface



The PSD9XX Functional Blocks (cont.)

9.3.3 Microcontroller Interface Examples

Figures 14 through 18 show examples of the basic connections between the PSD9XX and some popular microcontrollers. The PSD9XX Control input pins are labeled as to the microcontroller function for which they are configured. The MCU interface is specified using the PSDsoft.

9.3.3.1 80C31

Figure 14 shows the interface to the 80C31, which has an 8-bit multiplexed address/data bus. The lower address byte is multiplexed with the data bus. The microcontroller control signals PSEN, RD, and WR may be used for accessing the internal memory components and I/O Ports. The ALE input (pin PD0) latches the address.

9.3.3.2 80C251

The Intel 80C251 microcontroller features a user-configurable bus interface with four possible bus configurations, as shown in Table 19.

Configuration 1 is 80C31 compatible, and the bus interface to the PSD9XX is identical to that shown in Figure 14. Configurations 2 and 3 have the same bus connection as shown in Figure 15. There is only one read input (PSEN) connected to the CntI1 pin on the PSD9XX. The A16 connection to the PA0 pin allows for a larger address input to the PSD9XX. Configuration 4 is shown in Figure 16. The RD signal is connected to CntI1 and the PSEN signal is connected to the CNTL2.

The 80C251 has two major operating modes: Page Mode and Non-Page Mode. In Non-Page Mode, the data is multiplexed with the lower address byte, and ALE is active in every bus cycle. In Page Mode, data D[7:0] is multiplexed with address A[15:8]. In a bus cycle where there is a Page hit, the ALE signal is not active and only addresses A[7:0] are changing. The PSD9XX supports both modes. In Page Mode, the PSD bus timing is identical to Non-Page Mode except the address hold time and setup time with respect to ALE is not required. The PSD access time is measured from address A[7:0] valid to data in valid.

The PSD9XX Functional Blocks (cont.)

Table 19. 80C251 Configurations

| Configuration | 80C251 Read/Write Pins | Connecting to PSD9XX Pins | Page Mode |
|---|---|---|---|
| 1 | $\overline{WR}$<br>$\overline{RD}$<br>$\overline{PSEN}$ | CNTL0<br>CNTL1<br>CNTL2 | Non-Page Mode, 80C31 compatible<br>A[7:0] multiplex with D[7:0} |
| 2 | $\overline{WR}$<br>$\overline{PSEN}$ only | CNTL0<br>CNTL1 | Non-Page Mode<br>A[7:0] multiplex with D[7:0} |
| 3 | $\overline{WR}$<br>$\overline{PSEN}$ only | CNTL0<br>CNTL1 | Page Mode<br>A[15:8] multiplex with D[7:0} |
| 4 | $\overline{WR}$<br>$\overline{RD}$<br>$\overline{PSEN}$ | CNTL0<br>CNTL1<br>CNTL2 | Page Mode<br>A[15:8] multiplex with D[7:0} |

#### 9.3.3.3 80C51XA

The Philips 80C51XA microcontroller family supports an 8- or 16-bit multiplexed bus that can have burst cycles. Address bits A[3:0] are not multiplexed, while A[19:4] are multiplexed with data bits D[15:0] in 16-bit mode. In 8-bit mode, A[11:4] are multiplexed with data bits D[7:0].

The 80C51XA can be configured to operate in eight-bit data mode. (shown in Figure 17). The 80C51XA improves bus throughput and performance by executing Burst cycles for code fetches. In Burst Mode, address A19-4 are latched internally by the PSD9XX, while the 80C51XA changes the A3-0 lines to fetch up to 16 bytes of code. The PSD access time is then measured from address A3-A0 valid to data in valid. The PSD bus timing requirement in Burst Mode is identical to the normal bus cycle, except the address setup and hold time with respect to ALE does not apply.

#### 9.3.3.4 68HC11

Figure 18 shows an interface to a 68HC11 where the PSD9XX is configured in 8-bit multiplexed mode with E and R/W settings. The DPLD can generate the READ and WR signals for external devices.

Figure 14. Interfacing the PSD9XX with an 80C31

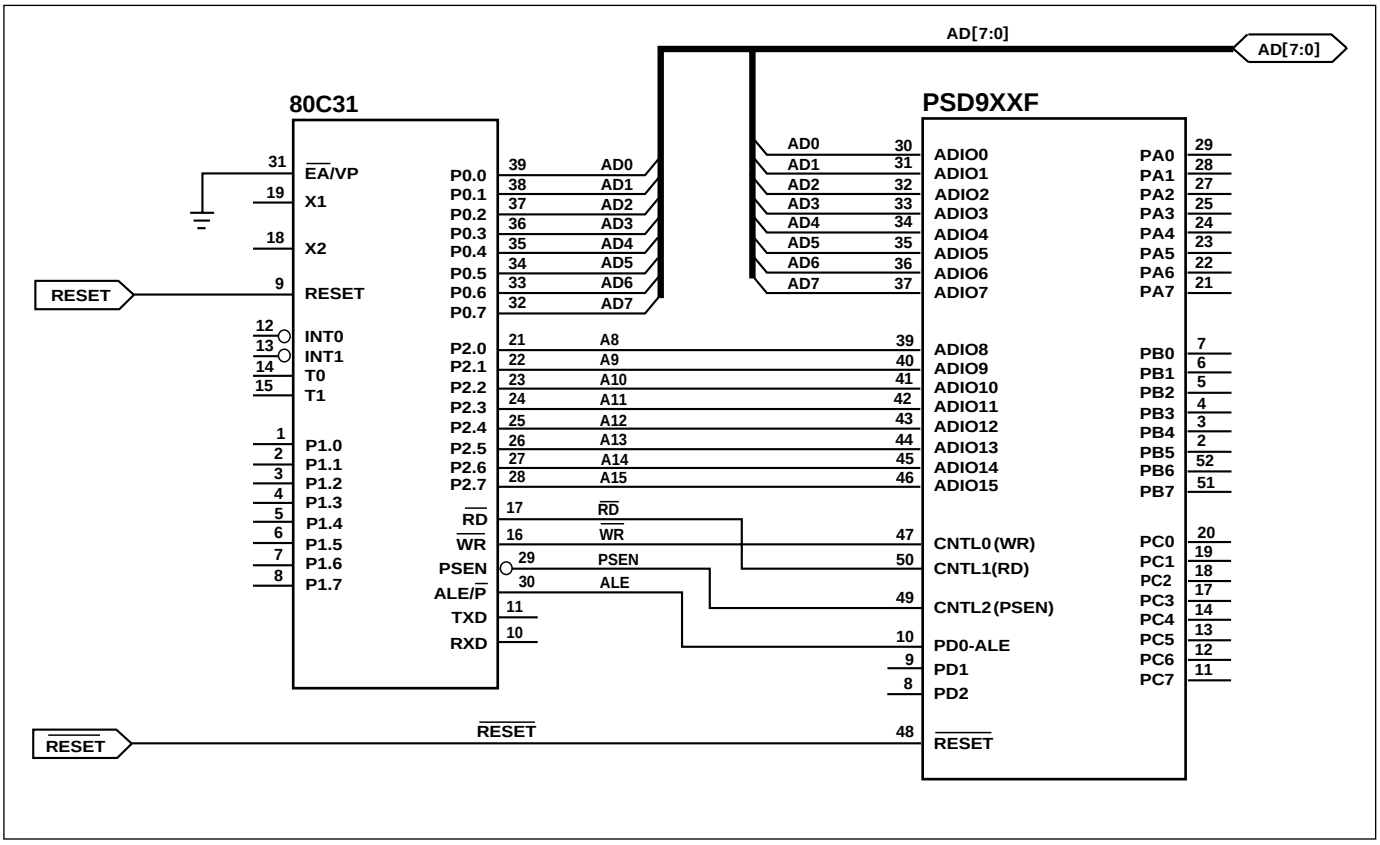
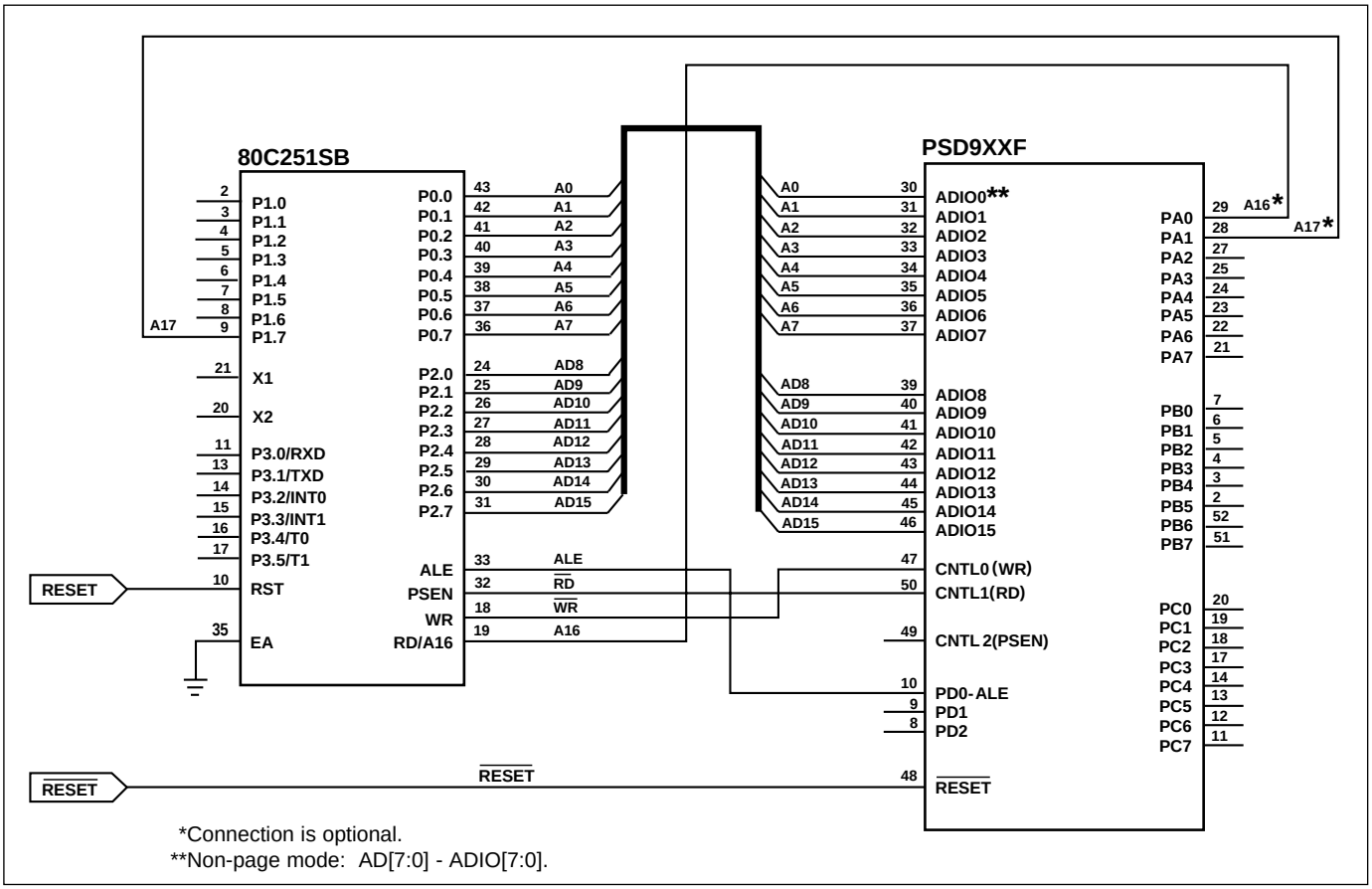



Figure 15. Interfacing the PSD9XX to the 80C251, with One Read Input




*Connection is optional.
**Non-page mode: AD[7:0] - ADIO[7:0].

Figure 16. Interfacing the PSD9XX to the 80C251, with Read and PSEN Inputs

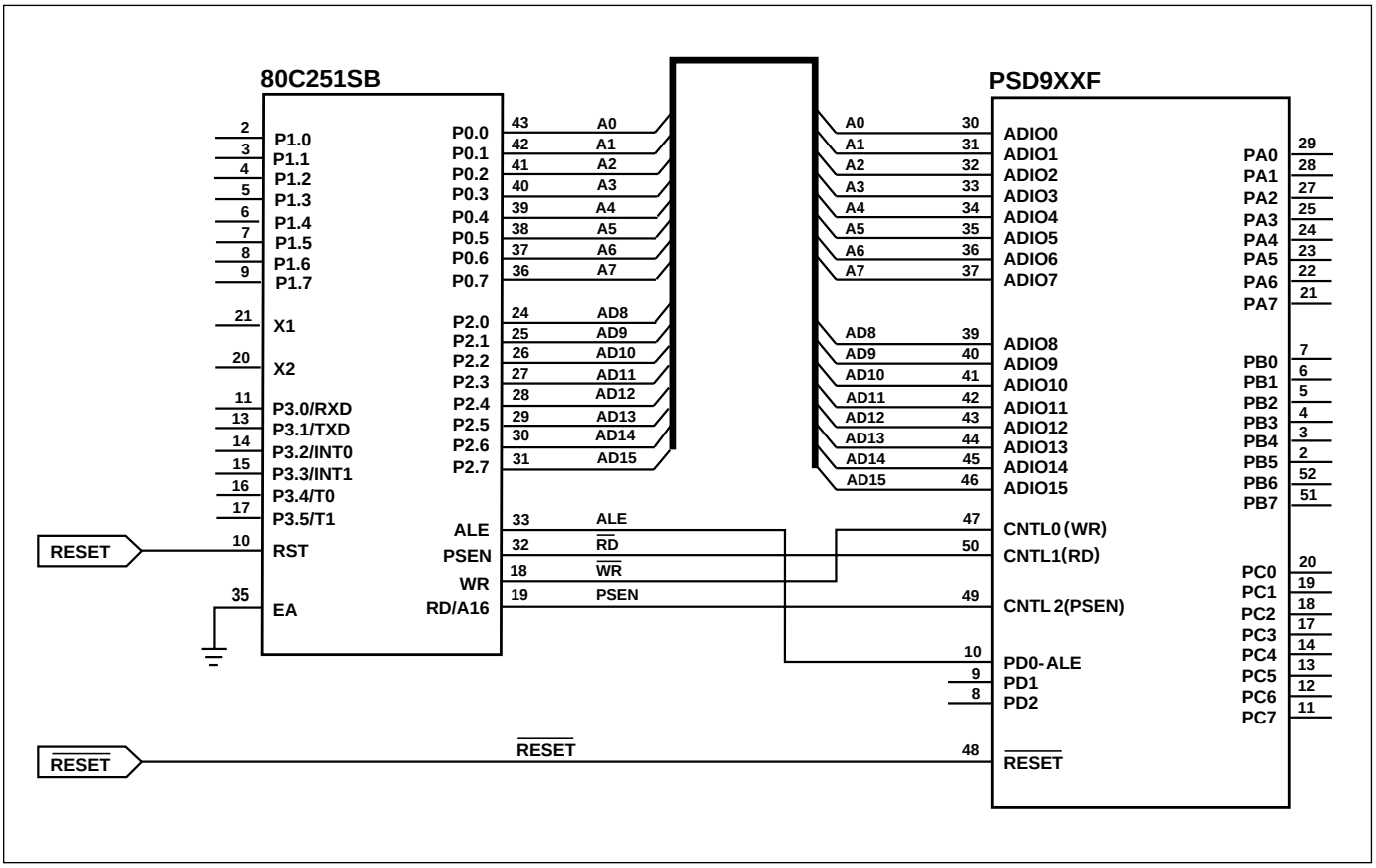



Figure 17. Interfacing the PSD9XX to the 80C51XA, 8-Bit Data Bus

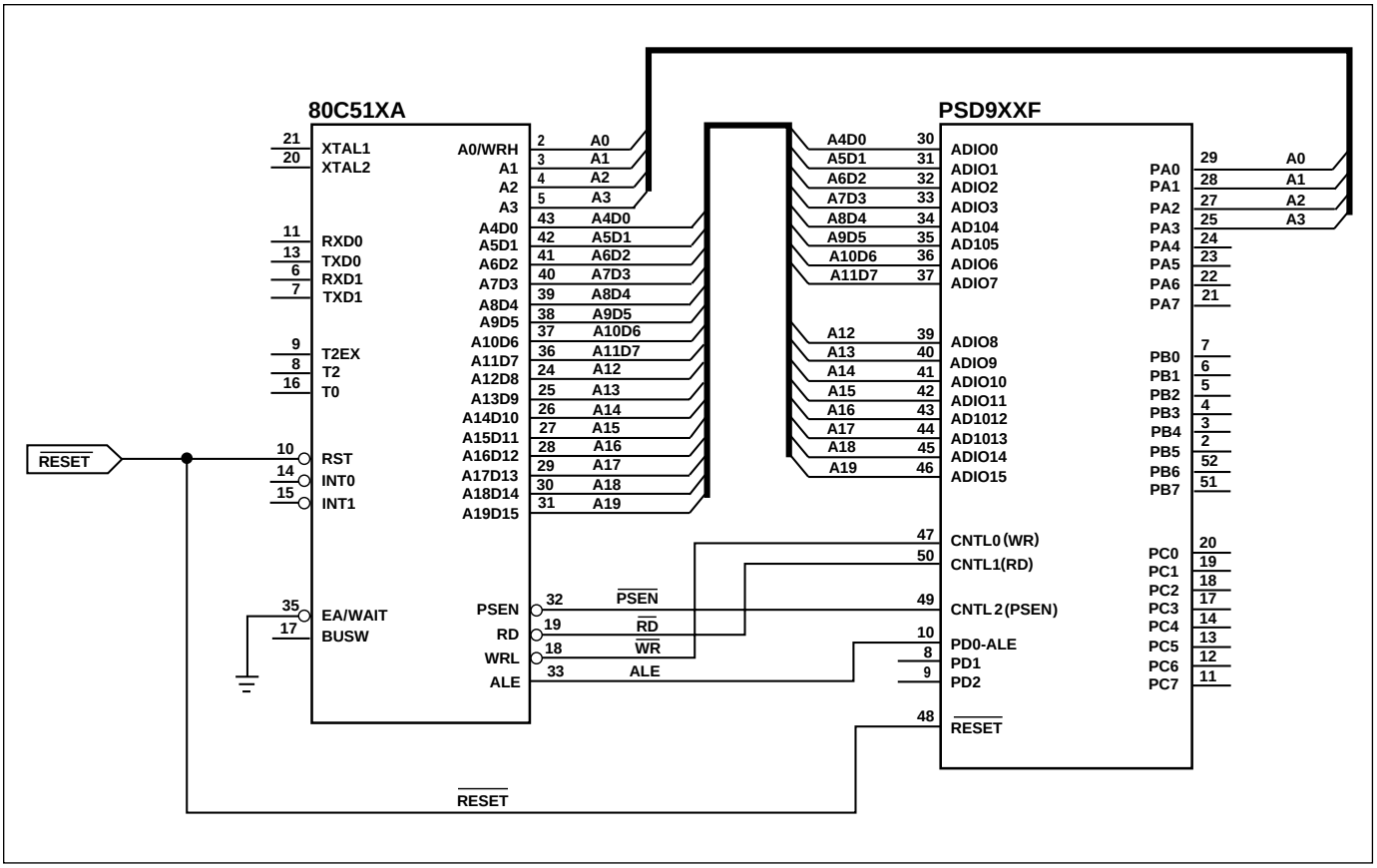
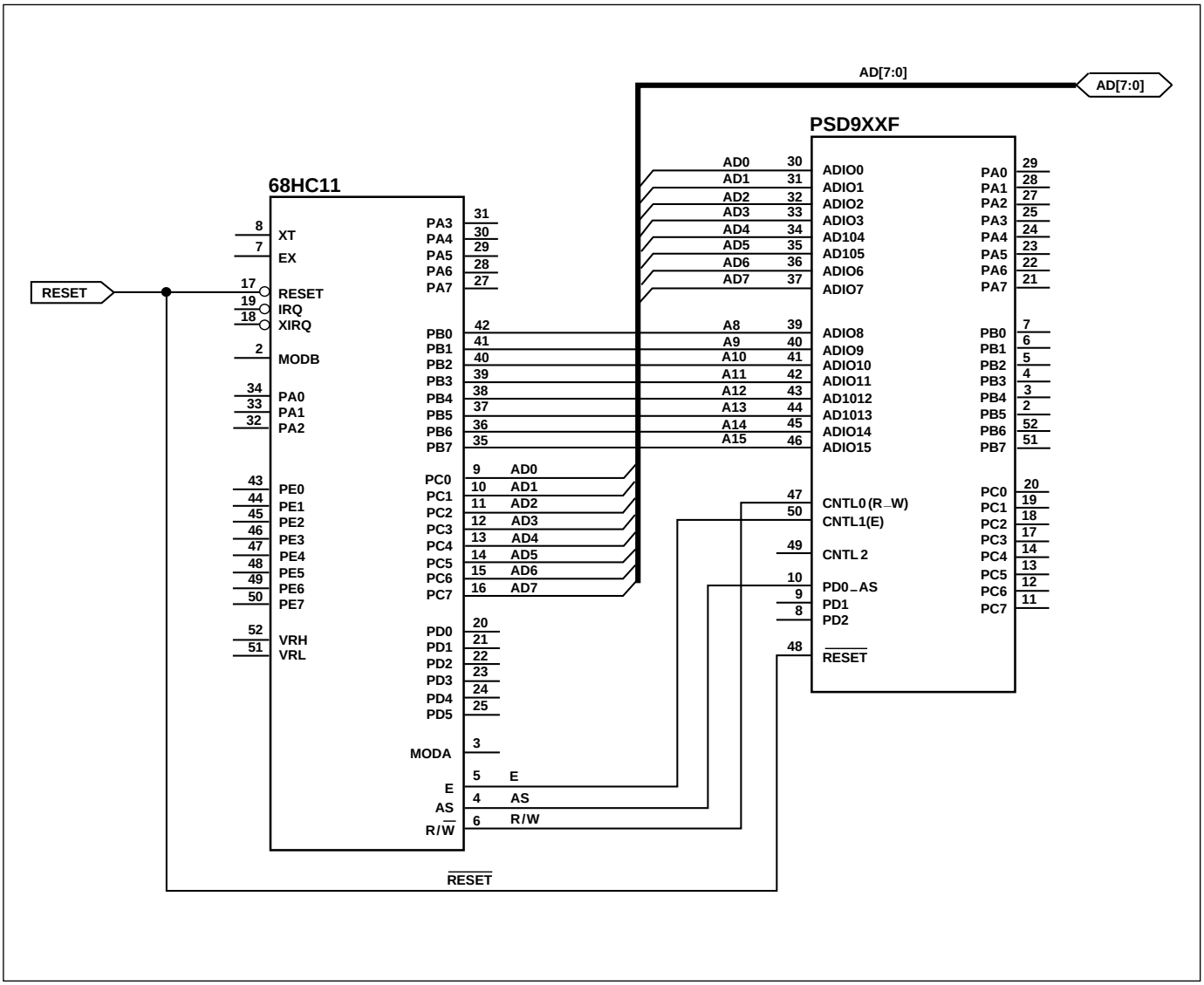


Figure 18. Interfacing the PSD9XX with a 68HC11 (Muxed Address/Data Bus)



The PSD9XX Functional Blocks (cont.)

## 9.4 I/O Ports

There are four programmable I/O ports: Ports A, B, C, and D. Each of the ports is eight bits except Port D, which is 3 bits. Each port pin is individually user configurable, thus allowing multiple functions per port. The ports are configured using PSDsoft or by the microcontroller writing to on-chip registers in the CSIOP address space.

The topics discussed in this section are:

- General Port Architecture
- Port Operating Modes
- Port Configuration Registers
- Port Data Registers
- Individual Port Functionality.

### 9.4.1 General Port Architecture

The general architecture of the I/O Port is shown in Figure 19. Individual Port architectures are shown in Figures 20 through 22. In general, once the purpose for a port pin has been defined, that pin will no longer be available for other purposes. Exceptions will be noted.

As shown in Figure 19, the ports contain an output multiplexer whose selects are driven by the configuration bits in the Control Registers (Ports A and B only) and PSDsoft. Inputs to the multiplexer include the following:

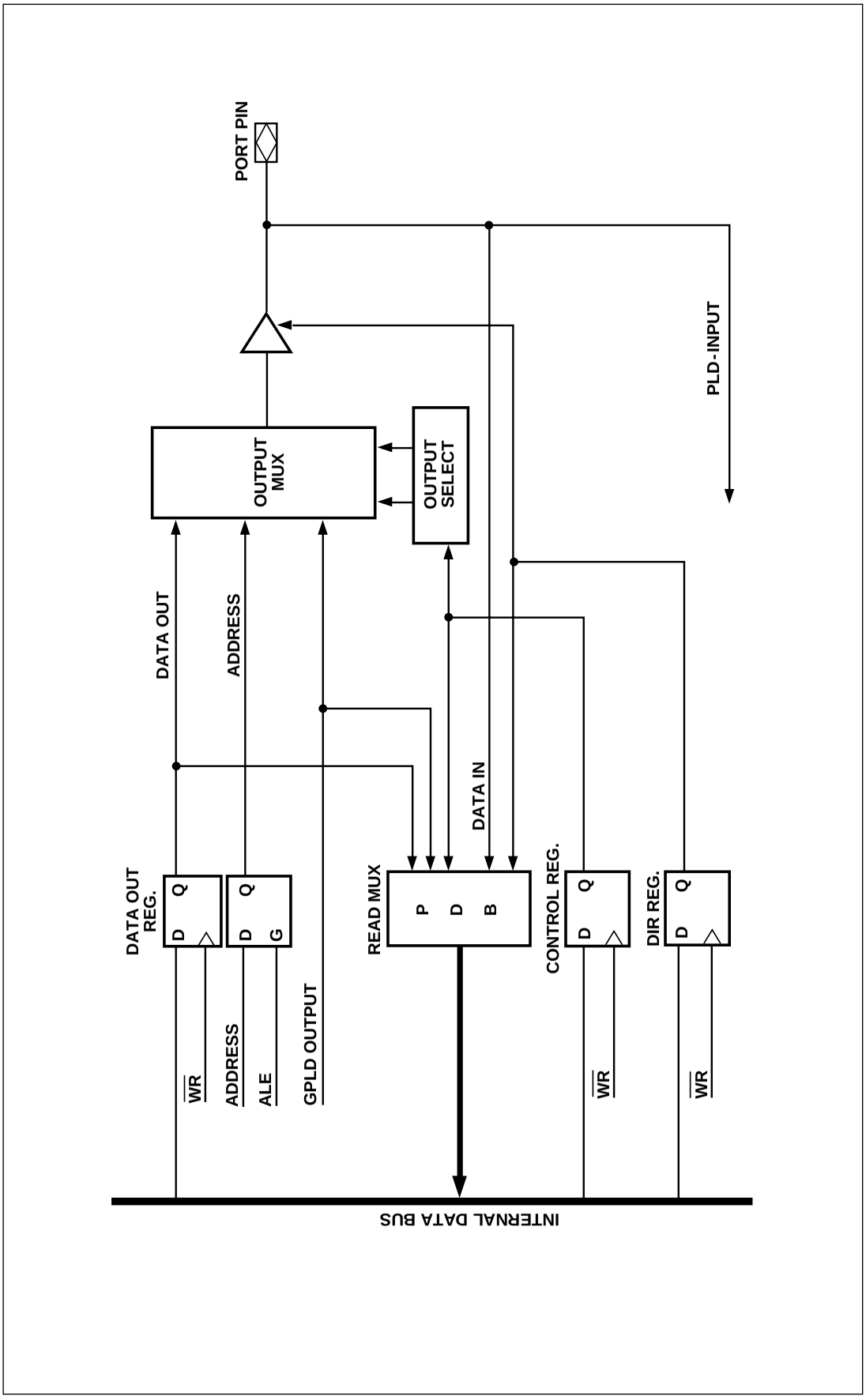
- ❏ Output data from the Data Out Register
- ❏ Latched address outputs
- ❏ General Purpose PLD (GPLD) outputs (external chip selects)

The Port Data Buffer (PDB) is a tri-state buffer that allows only one source at a time to be read. The PDB is connected to the Internal Data Bus for feedback and can be read by the microcontroller. The Data Out, Direction and Control Registers, and port pin input are all connected to the PDB.

The contents of these registers can be altered by the microcontroller. The PDB feedback path allows the microcontroller to check the contents of the registers.

The PSD9XX Functional Blocks (cont.)

Figure 19. General I/O Port Architecture



The PSD9XX Functional Blocks (cont.)

### 9.4.2 Port Operating Modes

The I/O Ports have several modes of operation. Some modes can be defined in PSDsoft, some by the microcontroller writing to the Control Registers in CSIOP space, and some by both. The modes that can only be defined using PSDsoft must be programmed into the device and cannot be changed unless the device is reprogrammed. The modes that can be changed by the microcontroller can be done so dynamically at run-time. The PLD I/O, Data Port, and Address Input, are the only modes that must be defined before programming the device. All other modes can be changed by the microcontroller at run-time.

Table 20 summarizes which modes are available on each port. Table 23 shows how and where the different modes are configured. Each of the port operating modes are described in the following subsections.

Table 20. Port Operating Modes

| Port Mode | Port A | Port B | Port C | Port D |
|---|---|---|---|---|
| MCU I/O | Yes | Yes | Yes | Yes |
| PLD Outputs | Yes | Yes | No | Yes |
| PLD Inputs | Yes | Yes | Yes | Yes |
| Address Out | Yes (A7 – 0) | Yes (A7 – 0) or A15 – 8) | No | No |
| Address In | Yes | Yes | Yes | Yes |
| Data Port | Yes (D7 – 0) | No | No | No |
| JTAG ISP | No | No | Yes | No |

The PSD9XX Functional Blocks (cont.)

Table 21. Port Operating Mode Settings

| Mode | Defined In PSDsoft | Control Register Setting at Run-Time | Direction Register Setting at Run-Time |
|---|---|---|---|
| MCU I/O | Declare pins only | 0 | 1 = output, 0 = input, |
| PLD I/O | Logic or chip select equations | NA | |
| Data Port (Port A) | Selected for MCU with non-mux bus | NA | NA |
| Address Out (Port A,B) | Declare pins only | 1 | 1 |
| Address In (Port A,B,C,D) | Declare pins only | NA | NA |
| JTAG ISP | Declare pins only | NA | NA |

*NA = Not Applicable

#### 9.4.2.1 MCU I/O Mode

In the MCU I/O Mode, the microcontroller uses the PSD9XX ports to expand its own I/O ports. By setting up the CSIOP space, the ports on the PSD9XX are mapped into the microcontroller address space. The addresses of the ports are listed in Table 7.

A port pin can be put into MCU I/O mode by writing a '0' to the corresponding bit in the Control Register. The MCU I/O direction may be changed by writing to the corresponding bit in the Direction Register. See the subsection on the Direction Register in the "Port Registers" section. When the pin is configured as an output, the content of the Data Out Register drives the pin. When configured as an input, the microcontroller can read the port input through the Data In buffer. See Figure 19.

Ports C and D do not have Control Registers, and are in MCU I/O mode by default. They can be used for PLD I/O if they are specified in PSDsoft.

#### 9.4.2.2 PLD I/O Mode

The PLD I/O Mode uses a port as an input to the PLDs, and/or as an output from the GPLD. The corresponding bit in the Direction Register must not be set to '1' if the pin is defined as a PLD input pin in PSDsoft. The PLD I/O Mode is specified in PSDsoft by declaring the port pins, and then specifying an equation in PSDsoft.

The PSD9XX Functional Blocks (cont.)

#### 9.4.2.3 Address Out Mode

For microcontrollers with a multiplexed address/data bus, Address Out Mode can be used to drive latched addresses onto the port pins. These port pins can, in turn, drive external devices. Either the output enable or the corresponding bits of both the Direction Register and Control Register must be set to a '1' for pins to use Address Out Mode. This must be done by the MCU at run-time. See Table 22 for the address output pin assignments on Ports A and B for various MCUs.

For non-multiplexed 8 bit bus mode, address lines A[7:0] are available to Port B in Address Out Mode.

**Note:** Do not drive address lines with Address Out Mode to an external memory device if it is intended for the MCU to boot from the external device. The MCU must first boot from PSD memory so the Direction and Control register bits can be set.

Table 22. I/O Port Latched Address Output Assignments

| Microcontroller | Port A (3:0) | Port A (7:4) | Port B (3:0) | Port B (7:4) |
|---|---|---|---|---|
| 8051XA (8-Bit) | N/A* | Address (7:4) | Address (11:8) | N/A |
| 80C251 (Page Mode) | N/A | N/A | Address (11:8) | Address (15:12) |
| All Other 8-Bit Multiplexed | Address (3:0) | Address (7:4) | Address (3:0) | Address (7:4) |
| 8-Bit Non-Multiplexed Bus | N/A | N/A | Address [3:0] | Address [7:4] |

N/A = Not Applicable.

#### 9.4.2.4 Address In Mode

For microcontrollers that have more than 16 address lines, the higher addresses can be connected to Port A, B, C, and D. The address input can be latched by the address strobe (ALE/AS). Any input that is included in the DPLD equations for the Main Flash, Secondary Flash, or SRAM is considered to be an address input.

#### 9.4.2.5 Data Port Mode

Port A can be used as a data bus port for a microcontroller with a non-multiplexed address/data bus. The Data Port is connected to the data bus of the microcontroller. The general I/O functions are disabled in Port A if the port is configured as a Data Port.

#### 9.4.2.6 JTAG ISP

Port C is JTAG compliant, and can be used for In-System Programming (ISP). For more information on the JTAG Port, refer to section 9.6.

The PSD9XX Functional Blocks (cont.)

#### 9.4.3 Port Configuration Registers (PCRs)

Each port has a set of PCRs used for configuration. The contents of the registers can be accessed by the microcontroller through normal read/write bus cycles at the addresses given in Table 7. The addresses in Table 7 are the offsets in hex from the base of the CSIOP register.

The pins of a port are individually configurable and each bit in the register controls its respective pin. For example, Bit 0 in a register refers to Bit 0 of its port. The three PCRs, shown in Table 23, are used for setting the port configurations. The default power-up state for each register in Table 23 is 00h.

Table 23. Port Configuration Registers

| Register Name | Port | MCU Access |
|---|---|---|
| Control | A,B | Write/Read |
| Direction | A,B,C,D | Write/Read |
| Drive Select* | A,B,C,D | Write/Read |

***NOTE:** See Table 27 for Drive Register bit definition.

##### 9.4.3.1 Control Register

Any bit set to '0' in the Control Register sets the corresponding Port pin to MCU I/O Mode, and a '1' sets it to Address Out Mode. The default mode is MCU I/O. Only Ports A and B have an associated Control Register.

##### 9.4.3.2 Direction Register

The Direction Register controls the direction of data flow in the I/O Ports. Any bit set to '1' in the Direction Register will cause the corresponding pin to be an output, and any bit set to '0' will cause it to be an input. The default mode for all port pins is input.

Figures 20 and 22 show the Port Architecture diagrams for Ports A, B and C, respectively. The direction of data flow for Ports A, B, and C are controlled by the direction register.

An example of a configuration for a port with the three least significant bits set to output and the remainder set to input is shown in Table 26. Since Port D only contains three pins, the Direction Register for Port D has only the three least significant bits active.

Table 24. Port Pin Direction Control

| Direction Register Bit | Port Pin Mode |
|---|---|
| 0 | Input |
| 1 | Output |

Table 26. Port Direction Assignment Example

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

The PSD9XX Functional Blocks (cont.)

9.4.3.3 Drive Select Register

The Drive Select Register configures the pin driver as Open Drain or CMOS for some port pins, and controls the slew rate for the other port pins. An external pull-up resistor should be used for pins configured as Open Drain.

A pin can be configured as Open Drain if its corresponding bit in the Drive Select Register is set to a '1'. The default pin drive is CMOS.

**Aside:** the slew rate is a measurement of the rise and fall times of an output. A higher slew rate means a faster output response and may create more electrical noise. A pin operates in a high slew rate when the corresponding bit in the Drive Register is set to '1'. The default rate is slow slew.

Table 27 shows the Drive Register for Ports A, B, C, and D. It summarizes which pins can be configured as Open Drain outputs and which pins the slew rate can be set for.

Table 27. Drive Register Pin Assignment

| Drive Register | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| Port A | Open Drain | Open Drain | Open Drain | Open Drain | Slew Rate | Slew Rate | Slew Rate | Slew Rate |
| Port B | Open Drain | Open Drain | Open Drain | Open Drain | Slew Rate | Slew Rate | Slew Rate | Slew Rate |
| Port D | NA | NA | NA | NA | NA | Slew Rate | Slew Rate | Slew Rate |

**NOTE:** NA = Not Applicable.

The PSD9XX Functional Blocks (cont.)

### 9.4.4 Port Data Registers

The Port Data Registers, shown in Table 28, are used by the microcontroller to write data to or read data from the ports. Table 28 shows the register name, the ports having each register type, and microcontroller access for each register type. The registers are described below.

#### 9.4.4.1 Data In

Port pins are connected directly to the Data In buffer. In MCU I/O input mode, the pin input is read through the Data In buffer.

#### 9.4.4.2 Data Out Register

Stores output data written by the MCU in the MCU I/O output mode. The contents of the Register are driven out to the pins if the Direction Register or the output enable product term is set to "1". The contents of the register can also be read back by the microcontroller.

Table 28. Port Data Registers

| Register Name | Port | MCU Access |
|---|---|---|
| Data In | A,B,C,D | Read – input on pin |
| Data Out | A,B,C,D | Write/Read |

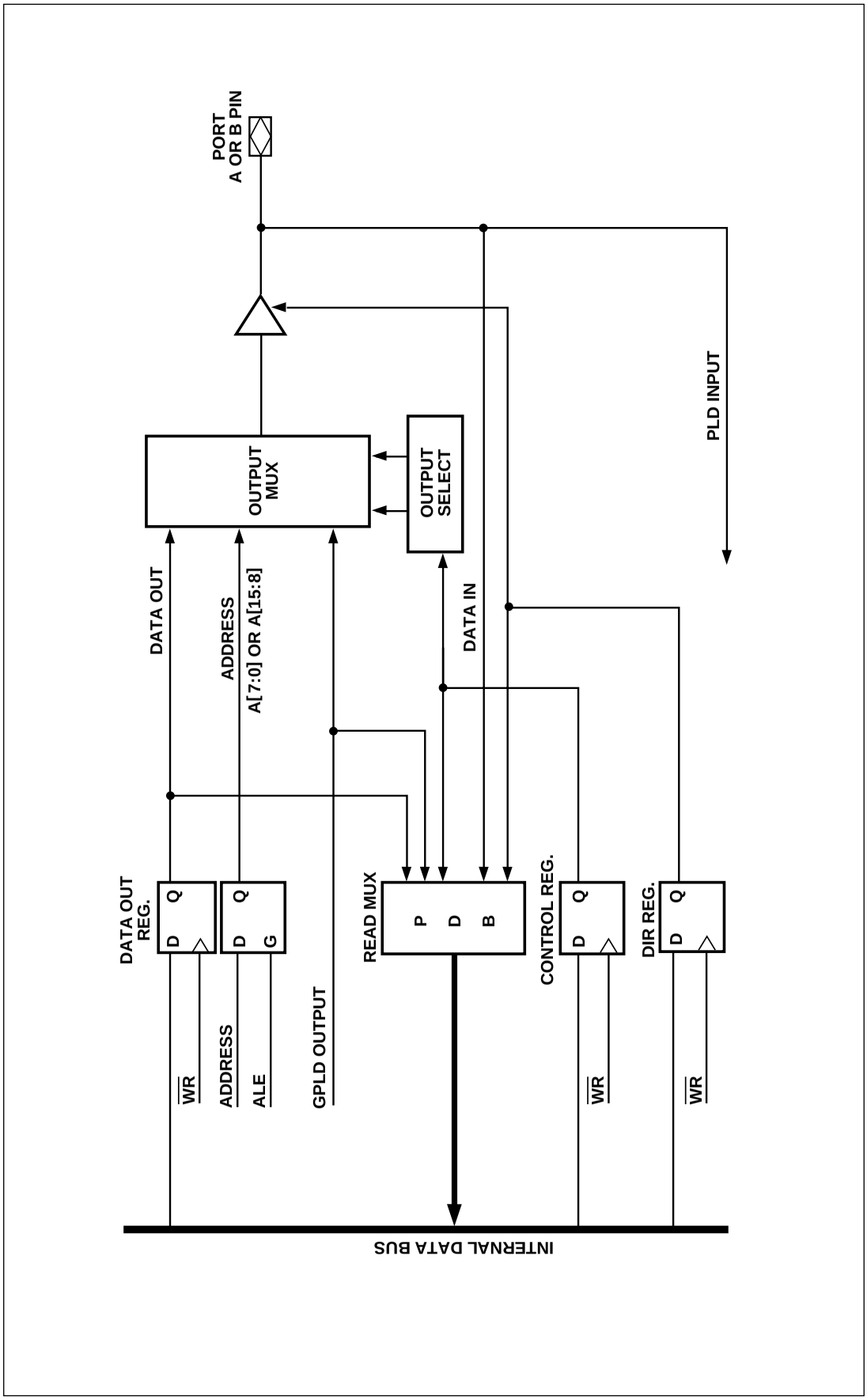
### 9.4.5 Ports A and B – Functionality and Structure

Ports A and B have similar functionality and structure, as shown in Figure 20. The two ports can be configured to perform one or more of the following functions:

- ❑ MCU I/O Mode
- ❑ GPLD Output – Combinatorial PLD outputs can be connected to Port A or Port B.
- ❑ PLD Input – Input to the PLDs.
- ❑ Latched Address output – Provide latched address output per Table 30.
- ❑ Address In – Additional high address inputs, may be latched by ALE.
- ❑ Open Drain/Slew Rate – pins PA[3:0] and PB[3:0] can be configured to fast slew rate, pins PA[7:4] and PB[7:4] can be configured to Open Drain Mode.
- ❑ Data Port – Port A only, connect to non-multiplexed 8-bit data bus.

## The PSD9XX Functional Blocks (cont.)

Figure 20. Ports A and B Structure



The PSD9XX Functional Blocks (cont.)

### 9.4.6 Port C – Functionality and Structure

Port C can be configured to perform one or more of the following functions (see Figure 21):

- ❑ MCU I/O Mode
- ❑ PLD Input – Input to the PLDs.
- ❑ Address In – Additional high address inputs using the Input Micro⇔Cells.
- ❑ In-System Programming – JTAG port can be enabled for programming/erase of the PSD9XX device. (See Section 9.6 for more information on JTAG programming.) Pins that are configured as JTAG pins in PSDsoft will not be available for other I/O functions.
- ❑ Open Drain – Port C pins can be configured in Open Drain Mode
- ❑ Battery Backup features – PC2 can be configured as a Battery Input (Vstby) pin. PC4 can be configured as a Battery On Indicator output pin, indicating when Vcc is less than Vbat.

Port C does not support Address Out mode, and therefore no Control Register is required. Pin PC7 may be configured as the DBE input in certain microcontroller interfaces.

### 9.4.7 Port D – Functionality and Structure

Port D has three I/O pins. See Figure 22. This port does not support Address Out mode, and therefore no Control Register is required. Port D can be configured to perform one or more of the following functions:

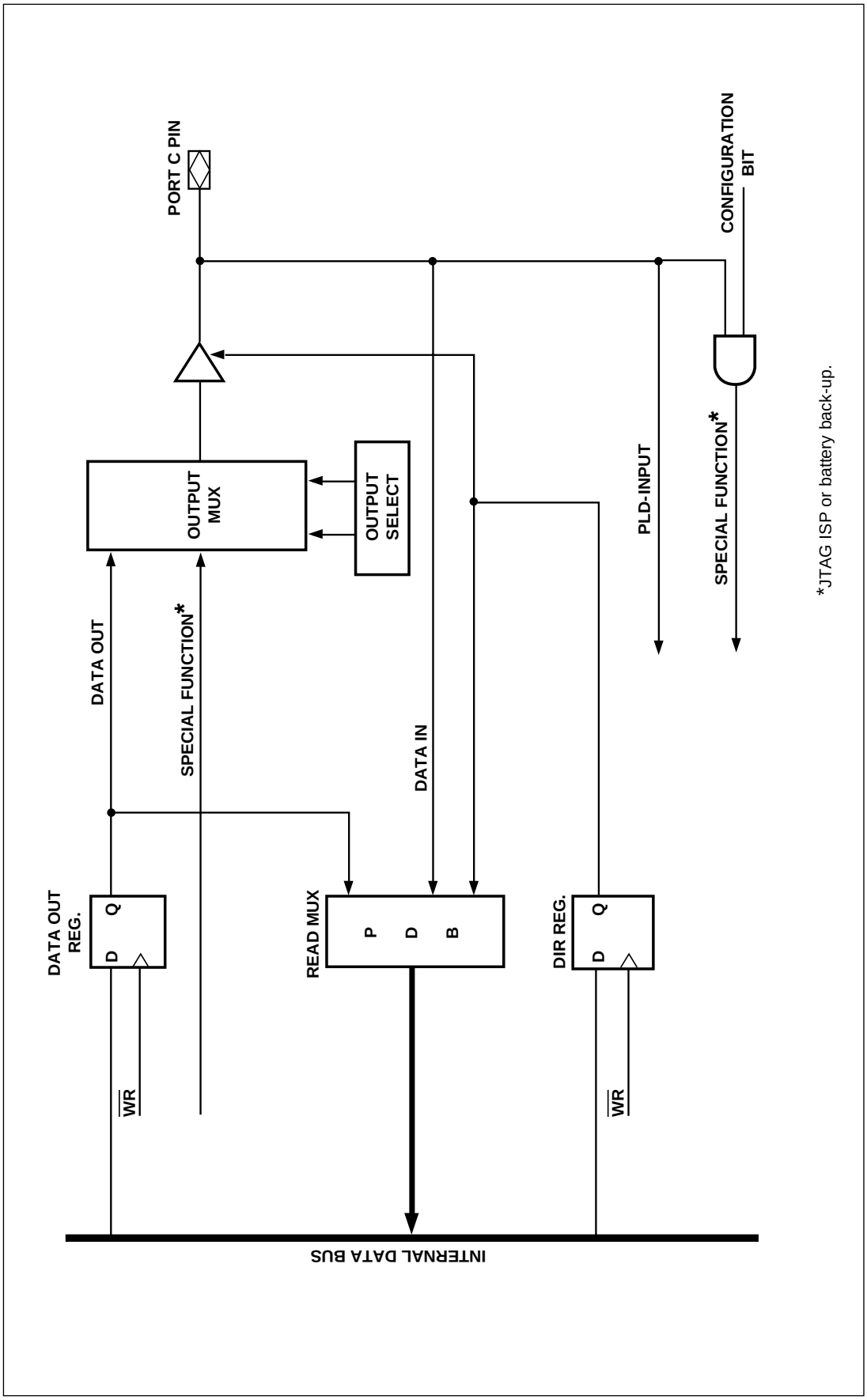
- ❑ MCU I/O Mode
- ❑ GPLD Output – Combinatorial PLD output (external chip selects)
- ❑ PLD Input – direct input to PLDs
- ❑ Slew rate – pins can be set up for fast slew rate

Port D pins can be configured in PSDsoft as input pins for other dedicated functions:

- ❑ PD0 – ALE, as address strobe input
- ❑ PD1 – CLKIN, as clock input to the PLD and APD counter
- ❑ PD2 – CSI, as active low chip select input. A high input will disable the Flash/SRAM and CSIOP.

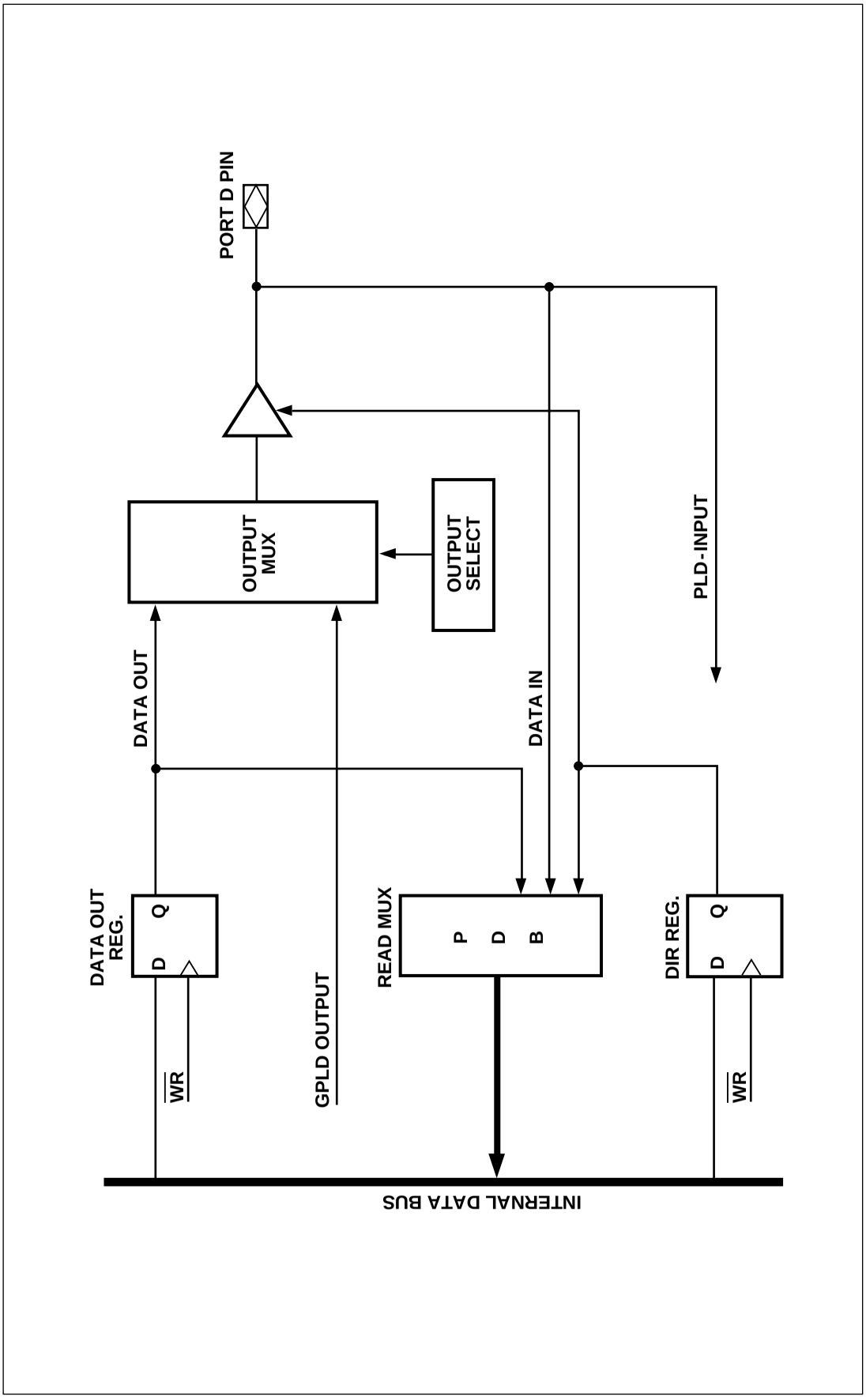
The PSD9XX Functional Blocks (cont.)

Figure 21. Port C Structure



The PSD9XX Functional Blocks (cont.)

Figure 22. Port D Structure



The PSD9XX Functional Blocks (cont.)

## 9.5 Power Management

The PSD9XX offers configurable power saving options. These options may be used individually or in combinations, as follows:

- All memory types in a PSD (Flash, Secondary Flash Block, and SRAM) are built with Zero-Power technology. In addition to using special silicon design methodology, Zero-Power technology puts the memories into standby mode when address/data inputs are not changing (zero DC current). As soon as a transition occurs on an input, the affected memory “wakes up”, changes and latches its outputs, then goes back to standby. The designer does **not** have to do anything special to achieve memory standby mode when no inputs are changing—it happens automatically.

  The PLD sections can also achieve standby mode when its inputs are not changing, see PMMR registers below.

- Like the Zero-Power feature, the Automatic Power Down (APD) logic allows the PSD to reduce to standby current automatically. The APD will block MCU address/data signals from reaching the memories and PLDs. This feature is available on all PSD9XX devices. The APD unit is described in more detail in section 9.5.1.

  Built in logic will monitor the address strobe of the MCU for activity. If there is no activity for a certain time period (MCU is asleep), the APD logic initiates Power Down Mode (if enabled). Once in Power Down Mode, all address/data signals are blocked from reaching PSD memories and PLDs, and the memories are deselected internally. This allows the memories and PLDs to remain in standby mode even if the address/data lines are changing state externally (noise, other devices on the MCU bus, etc.). Keep in mind that any unblocked PLD input signals (not MCU address) that are changing states keeps the PLD out of standby mode, but not the memories.

- The PSD Chip Select Input (CSI) on all families can be used to disable the internal memories, placing them in standby mode even if inputs are changing. This feature does not block any internal signals or disable the PLDs. This is a good alternative to using the APD logic, especially if your MCU has a chip select output. There is a slight penalty in memory access time when the CSI signal makes its initial transition from deselected to selected.

- The PMMR registers can be written by the MCU at run-time to manage power. All PSD devices support “blocking bits” in these registers that are set to block designated signals from reaching both PLDs. Current consumption of the PLDs is directly related to the composite frequency of the changes on their inputs (see Figure 26). Significant power savings can be achieved by blocking signals that are not used in PLD equations.

  The PSD9XX devices have a Turbo Bit in the PMMR0 register. This bit can be set to disable the Turbo Mode feature (default is Turbo Mode on). While Turbo Mode is disabled, the PLDs can achieve standby current when no PLD inputs are changing (zero DC current). Even when inputs do change, significant power can be saved at lower frequencies (AC current), compared to when Turbo Mode is enabled. Conversely, when the Turbo Mode is enabled, there is a significant DC current component and the AC component is higher.

### 9.5.1 Automatic Power Down (APD) Unit and Power Down Mode

The APD Unit, shown in Figure 23, puts the PSD into Power Down Mode by monitoring the activity of the address strobe (ALE/AS). If the APD unit is enabled, as soon as activity on the address strobe stops, a four bit counter starts counting. If the address strobe remains inactive for fifteen clock periods of the CLKIN signal, the Power Down (PDN) signal becomes active, and the PSD will enter into Power Down Mode, discussed next.

The PSD9XX Functional Blocks (cont.)

9.5.1 Automatic Power Down (APD) Unit and Power Down Mode (cont.)

Power Down Mode

By default, if you enable the PSD APD unit, Power Down Mode is automatically enabled. The device will enter Power Down Mode if the address strobe (ALE/AS) remains inactive for fifteen CLKIN (pin PD1) clock periods.

The following should be kept in mind when the PSD is in Power Down Mode:

- If the address strobe starts pulsing again, the PSD will return to normal operation. The PSD will also return to normal operation if either the CSI input returns low or the Reset input returns high.
- The MCU address/data bus is blocked from all memories and PLDs.
- Various signals can be blocked (prior to Power Down Mode) from entering the PLDs by setting the appropriate bits in the PMMR registers. The blocked signals include MCU control signals and the common clock (CLKIN). Note that blocking CLKIN from the PLDs will not block CLKIN from the APD unit.
- All PSD memories enter Standby Mode and are drawing standby current. However, the PLDs and I/O ports do **not** go into Standby Mode because you don't want to have to wait for the logic and I/O to "wake-up" before their outputs can change. See table 29 for Power Down Mode effects on PSD ports.
- Typical standby current is in µA for 5 V parts. This standby current value assumes that there are no transitions on any PLD input.

Table 29. Power Down Mode's Effect on Ports

| Port Function | Pin Level |
|---|---|
| MCU I/O | No Change |
| PLD Out | No Change |
| Address Out | Undefined |
| Data Port | Three-State |

Table 30. PSD9XX Timing and Standby Current During Power Down Mode

| Mode | PLD Propagation Delay | Memory Access Time | Access Recovery Time to Normal Access | 5V $V_{CC}$, Typical Standby Current |
|---|---|---|---|---|
| Power Down | Normal tpd (Note 1) | No Access | tLVDV | 75 µA (Note 2) |

**NOTES:** 1. Power Down does not affect the operation of the PLD. The PLD operation in this mode is based only on the Turbo Bit.
2. Typical current consumption assuming no PLD inputs are changing state and the PLD Turbo bit is off.

**HC11 (or compatible) Users Note**

The HC11 turns off its E clock when it sleeps. Therefore, if you are using an HC11 (or compatible) in your design, and you wish to use the Power Down, you must not connect the E clock to the CLKIN input (PD1). You should instead connect an independent clock signal to the CLKIN input. The clock frequency must be **less than** 15 times the frequency of AS. The reason for this is that if the frequency is greater than 15 times the frequency of AS, the PSD9XX will keep going into Power Down Mode.

The PSD9XX Functional Blocks (cont.)

Figure 23. APD Logic Block

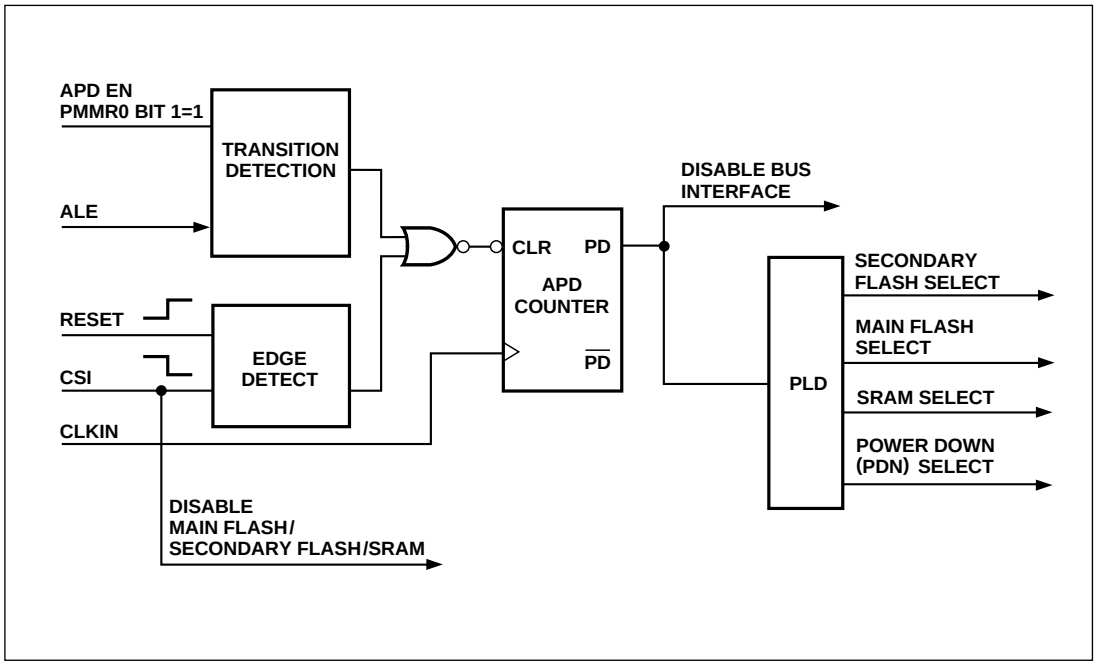
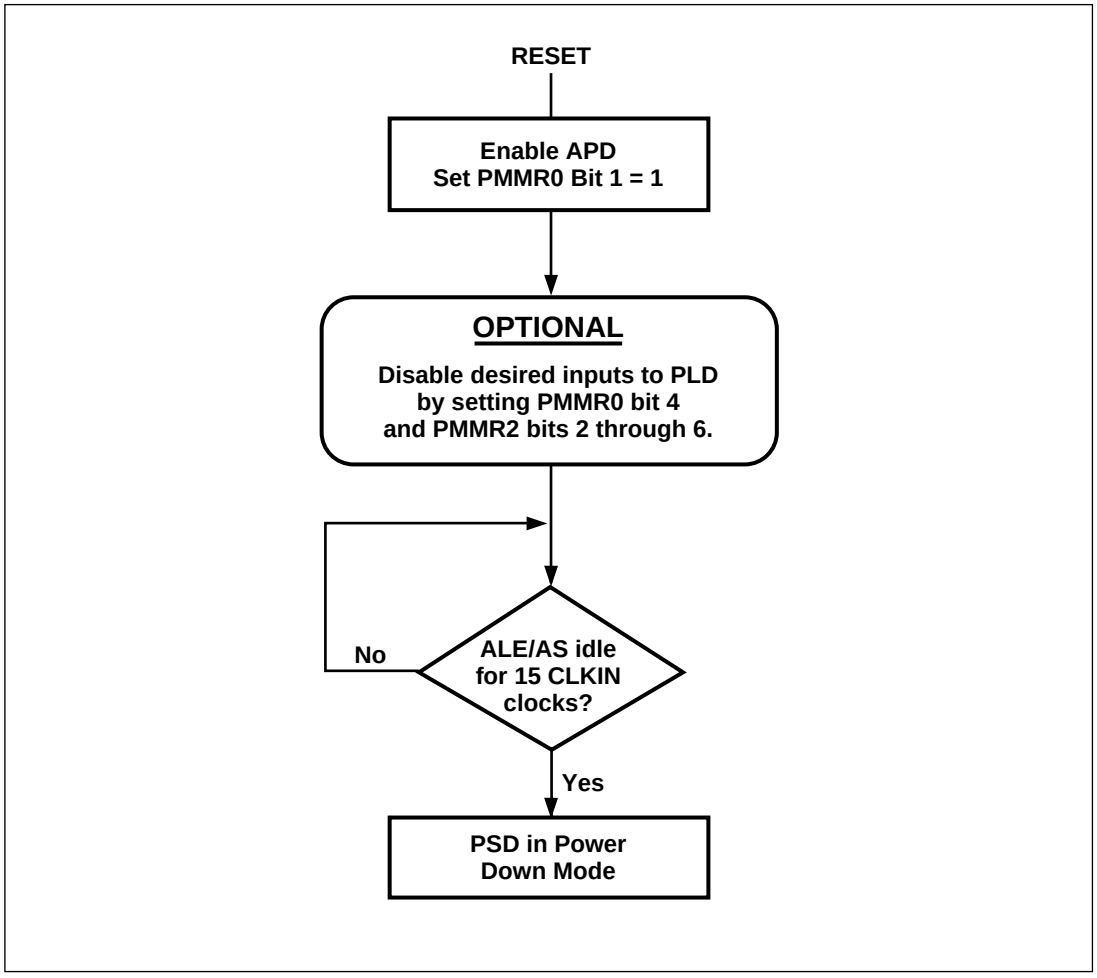



Figure 24. Enable Power Down Flow Chart



## The PSD9XX Functional Blocks (cont.)

Table 31. Power Management Mode Registers (PMMR0, PMMR2)**

PMMR0

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| * | * | * | PLD Array clk | PLD Turbo | * | APD Enable | * |
| | | | 1 = off | 1 = off | | 1 = on | |

*Bits 0, 2, 6, and 7 are not used, and should be set to 0, bit 5 should be set to 1.

**The PMMR0, and PMMR2 register bits are cleared to zero following power up. Subsequent reset pulses will not clear the registers.

Bit 1 0 = Automatic Power Down (APD) is disabled.
1 = Automatic Power Down (APD) is enabled.

Bit 3 0 = PLD Turbo is on.
1 = PLD Turbo is off, saving power.

Bit 4 0 = CLKIN input to the PLD AND array is connected.
Every CLKIN change will power up the PLD when Turbo bit is off.
1 = CLKIN input to PLD AND array is disconnected, saving power.

PMMR2

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|
| * | PLD array DBE | PLD array ALE | PLD** array CNTL2 | PLD** array CNTL1 | PLD** array CNTL0 | * | * |
| | 1 = off | 1 = off | 1 = off | 1 = off | 1 = off | | |

*Unused bits should be set to 0.

**Refer to Table 17 the signals that are blocked on pins CNTL0-2.

Bit 2 0 = Cntl0 input to the PLD AND array is connected.
1 = Cntl0 input to PLD AND array is disconnected, saving power.

Bit 3 0 = Cntl1 input to the PLD AND array is connected.
1 = Cntl1 input to PLD AND array is disconnected, saving power.

Bit 4 0 = Cntl2 input to the PLD AND array is connected.
1 = Cntl2 input to PLD AND array is disconnected, saving power.

Bit 5 0 = ALE input to the PLD AND array is connected.
1 = ALE input to PLD AND array is disconnected, saving power.

Bit 6 0 = DBE input to the PLD AND array is connected.
1 = DBE input to PLD AND array is disconnected, saving power.

The PSD9XX Functional Blocks (cont.)

Table 32. APD Counter Operation

| APD Enable Bit | ALE Power Down Polarity | ALE Level | APD Counter |
|---|---|---|---|
| 0 | X | X | Not Counting |
| 1 | X | Pulsing | Not Counting |
| 1 | 1 | 1 | Counting (Generates PDN after 15 Clocks) |
| 1 | 0 | 0 | Counting (Generates PDN after 15 Clocks) |

### 9.5.2 Other Power Saving Options

The PSD9XX offers other reduced power saving options that are independent of the Power Down Mode. Except for the SRAM Standby and CSI input features, they are enabled by setting bits in the PMMR0 and PMMR2 registers.

#### 9.5.2.1 Zero Power PLD

The power and speed of the PLDs are controlled by the Turbo bit (bit 3) in the PMMR0. By setting the bit to “1”, the Turbo mode is disabled and the PLDs consume Zero Power current when the inputs are not switching for an extended time of 70 ns. The propagation delay time will be increased by 10 ns after the Turbo bit is set to “1” (turned off) when the inputs change at a composite frequency of less than 15 MHz. When the Turbo bit is set to a “0” (turned on), the PLDs run at full power and speed. The Turbo bit affects the PLD’s D.C. power, AC power, and propagation delay.

**Note:** Blocking MCU control signals with PMMR2 bits can further reduce PLD AC power consumption.

#### 9.5.2.2 SRAM Standby Mode (Battery Backup)

The PSD9XX supports a battery backup operation that retains the contents of the SRAM in the event of a power loss. The SRAM has a Vstby pin (PC2) that can be connected to an external battery. When $V_{CC}$ becomes lower than Vstby then the PSD will automatically connect to Vstby as a power source to the SRAM. The SRAM Standby Current (Istby) is typically 0.5 µA. The SRAM data retention voltage is 2 V minimum. The battery-on indicator (Vbaton) can be routed to PC4. This signal indicates when the $V_{CC}$ has dropped below the Vstby voltage, and that the SRAM is running on battery power.

#### 9.5.2.3 The CSI Input

Pin PD2 of Port D can be configured in PSDsoft as the CSI input. When low, the signal selects and enables the internal Flash, Boot Block, SRAM, and I/O for read or write operations involving the PSD9XX. A high on the CSI pin will disable the Flash memory, Boot Block, and SRAM, and reduce the PSD power consumption. However, the PLD and I/O pins remain operational when CSI is high. **Note:** there may be a timing penalty when using the CSI pin depending on the speed grade of the PSD that you are using. See the timing parameter $t_{SLQV}$ in the AC/DC specs.

#### 9.5.2.4 Input Clock

The PSD9XX provides the option to turn off the CLKIN input to the PLD AND array to save AC power consumption. During Power Down Mode, or, if the CLKIN input is not being used as part of the PLD logic equation, the clock should be disabled to save AC power. The CLKIN will be disconnected from the PLD AND array setting bit 4 to a “1” in PMMR0.

#### 9.5.2.5 MCU Control Signals

The PSD9XX provides the option to turn off the input control signals (CNTL0-2, ALE, and DBE) to the PLD to save AC power consumption. These control signals are inputs to the PLD AND array. During Power Down Mode, or, if any of them are not being used as part of the PLD logic equation, these control signals should be disabled to save AC power. They will be disconnected from the PLD AND array by setting bits 2, 3, 4, 5, and 6 to a “1” in the PMMR2. Note that blocking MCU control signals to the GPLD will not block these signals from reaching the memory and I/O sections of the chip.

The PSD9XX Functional Blocks (cont.)

### 9.5.3 Reset and Power On Requirement

#### 9.5.3.1 Power On Reset

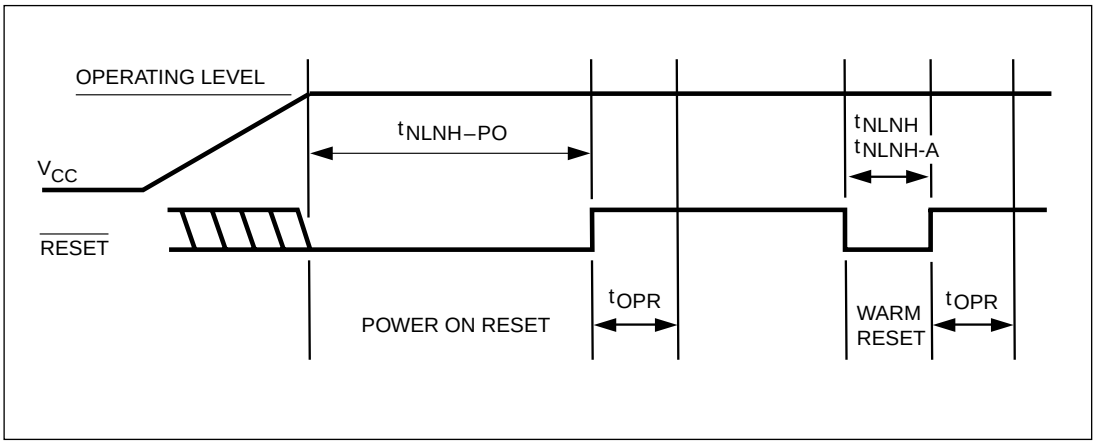
Upon power up the PSD9XX requires a reset pulse of tNLNH-PO (minimum 1 ms) after $V_{CC}$ is steady. During this time period the device loads internal configurations, clears some of the registers and sets the Flash into operating mode. After the rising edge of reset, the PSD9XX remains in the reset state for an additional tOPR (maximum 120 ns) nanoseconds before the first memory access is allowed.

The PSD9XX Flash memory is reset to the read array mode upon power up. The FSi and CSBOOTi select signals along with the write strobe signal must be in the false state during power-up reset for maximum security of the data contents and to remove the possibility of a byte being written on the first edge of a write strobe signal. Any Flash memory write cycle initiation is prevented automatically when $V_{CC}$ is below VLKO.

#### 9.5.3.2 Warm Reset

Once the device is up and running, the device can be reset with a much shorter pulse of tNLNH (minimum 150 ns). The same tOPR time is needed before the device is operational after warm reset. Figure 25 shows the timing of the power on and warm reset.

Figure 25. Power On and Warm Reset Timing




#### 9.5.3.3 I/O Pin, Register and PLD Status at Reset

Table 33 shows the I/O pin, register and PLD status during power on reset, warm reset and power down mode. PLD outputs are always valid during warm reset, and they are valid in power on reset once the internal PSD configuration bits are loaded. This loading of PSD is completed typically long before the $V_{CC}$ ramps up to operating level. Once the PLD is active, the state of the outputs are determined by the PLD equations.

The PSD9XX Functional Blocks (cont.)

Table 33. Status During Power On Reset, Warm Reset and Power Down Mode

| Port Configuration | Power On Reset | Warm Reset | Power Down Mode |
|---|---|---|---|
| MCU I/O | Input Mode | Input Mode | Unchanged |
| PLD Output | Valid after internal PSD configuration bits are loaded | Valid | Depend on inputs to PLD (address are blocked in PD mode) |
| Address Out | Tri-stated | Tri-stated | Not defined |
| Data Port | Tri-stated | Tri-stated | Tri-stated |

| Register | Power On Reset | Warm Reset | Power Down Mode |
|---|---|---|---|
| PMMR0, 2 | Cleared to “0” | Unchanged | Unchanged |
| VM Register* | Initialized based on the selection in PSDsoft Configuration Menu. | Initialized based on the selection in PSDsoft Configuration Menu. | Unchanged |
| All other registers | Cleared to “0” | Cleared to “0” | Unchanged |

*SR_cod bit in the VM Register are always cleared to zero on power on or warm reset.

9.5.3.4 Reset of Flash Erase and Programming Cycles (PSD934F2 and PSD954F2)
An external reset on the RESET pin will also reset the internal Flash memory state machine. When the Flash is in programming or erase mode, the RESET pin will terminate the programming or erase operation and return the Flash back to read mode in tNLNH-A (minimum 25 µs) time.

## 9.6 Programming In-Circuit using the JTAG Interface

The JTAG interface on the PSD9XX can be enabled on Port C (see Table 34). All memory (Flash and Secondary Flash Block), PLD logic, and PSD configuration bits may be programmed through the JTAG interface. A blank part can be mounted on a printed circuit board and programmed using JTAG.

The standard JTAG signals (IEEE 1149.1) are TMS, TCK, TDI, and TDO. Two additional signals, $\overline{\text{TSTAT}}$ and $\overline{\text{TERR}}$, are optional JTAG extensions used to speed up program and erase operations.

By default, on a blank PSD (as shipped from factory or after erasure), four pins on Port C are enabled for the basic JTAG signals TMS, TCK, TDI, and TDO.

See Application Note 54 for more details on JTAG In-System-Programming.

Table 34. JTAG Port Signals

| Port C Pin | JTAG Signals | Description |
|---|---|---|
| PC0 | TMS | Mode Select |
| PC1 | TCK | Clock |
| PC3 | $\overline{\text{TSTAT}}$ | Status |
| PC4 | $\overline{\text{TERR}}$ | Error Flag |
| PC5 | TDI | Serial Data In |
| PC6 | TDO | Serial Data Out |

The PSD9XX Functional Blocks (cont.)

### 9.6.1 Standard JTAG Signals

The JTAG configuration bit (non-volatile) inside the PSD can be set by the user in the PSDsoft. Once this bit is set and programmed in the PSD, the JTAG pins are dedicated to JTAG at all times and is in compliance with IEEE 1149.1. After power up the standard JTAG signals (TDI, TDO TCK and TMS) are inputs, waiting for a serial command from an external JTAG controller device (such as FlashLink or Automated Test Equipment). When the enabling command is received from the external JTAG controller, TDO becomes an output and the JTAG channel is fully functional inside the PSD. The same command that enables the JTAG channel may optionally enable the two additional JTAG pins, TSTAT and TERR.

The PSD9XX supports JTAG In-System-Configuration (ISC) commands, but not Boundary Scan. ST's PSDsoft software tool and FlashLink JTAG programming cable implement these JTAG-ISC commands.

### 9.6.2 JTAG Extensions

$\overline{\text{TSTAT}}$ and $\overline{\text{TERR}}$ are two JTAG extension signals enabled by a JTAG command received over the four standard JTAG pins (TMS, TCK, TDI, and TDO). They are used to speed programming and erase functions by indicating status on PSD pins instead of having to scan the status out serially using the standard JTAG channel. See Application Note 54.

$\overline{\text{TERR}}$ will indicate if an error has occurred when erasing a sector or programming a byte in Flash memory. This signal will go low (active) when an error condition occurs, and stay low until a special JTAG command is executed or a chip reset pulse is received after an "ISC-DISABLE" command.

$\overline{\text{TSTAT}}$ behaves the same as the Rdy/Bsy signal described in section 9.1.1.2. $\overline{\text{TSTAT}}$ will be high when the PSD9XX device is in read array mode (Flash memory and Boot Block contents can be read). $\overline{\text{TSTAT}}$ will be low when Flash memory programming or erase cycles are in progress, and also when data is being written to the Secondary Flash Block.

$\overline{\text{TSTAT}}$ and $\overline{\text{TERR}}$ can be configured as open-drain type signals with a JTAG command.

### 9.6.3 Security and Flash Memories Protection

When the security bit is set, the device cannot be read on a device programmer or through the JTAG Port. When using the JTAG Port, only a full chip erase command is allowed. All other program/erase/verify commands are blocked. Full chip erase returns the part to a non-secured blank state. The Security Bit can be set in PSDsoft.

All Flash Memory and Boot sectors can individually be sector protected against erasures. The sector protect bits can be set in PSDsoft.

## Absolute Maximum Ratings

| Symbol | Parameter | Condition | Min | Max | Unit |
|---|---|---|---|---|---|
| $T_{STG}$ | Storage Temperature | PLDCC | – 65 | + 125 | °C |
| | Operating Temperature | Commercial | 0 | + 70 | °C |
| | | Industrial | – 40 | + 85 | °C |
| | Voltage on any Pin | With Respect to GND | – 0.6 | + 7 | V |
| $V_{PP}$ | Device Programmer Supply Voltage | With Respect to GND | – 0.6 | + 14 | V |
| $V_{CC}$ | Supply Voltage | With Respect to GND | – 0.6 | + 7 | V |
| | ESD Protection | | >2000 | | V |

**NOTE:** Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not recommended. Exposure to Absolute Maximum Rating conditions for extended periods of time may affect device reliability.

## Operating Range

| Range | Temperature | $V_{CC}$ Tolerance |
|---|---|---|
| Commercial | 0° C to +70°C | + 5 V ± 10% |
| Industrial | –40° C to +85°C | + 5 V ± 10% |
| Commercial | 0° C to +70°C | 3.0 V to 3.6 V |
| Industrial | –40° C to +85°C | 3.0 V to 3.6 V |

## Recommended Operating Conditions

| Symbol | Parameter | Condition | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $V_{CC}$ | Supply Voltage | All Speeds | 4.5 | 5 | 5.5 | V |
| $V_{CC}$ | Supply Voltage | V-Versions All Speeds | 3.0 V to 3.6 V | | 3.6 | V |

## AC/DC Parameters

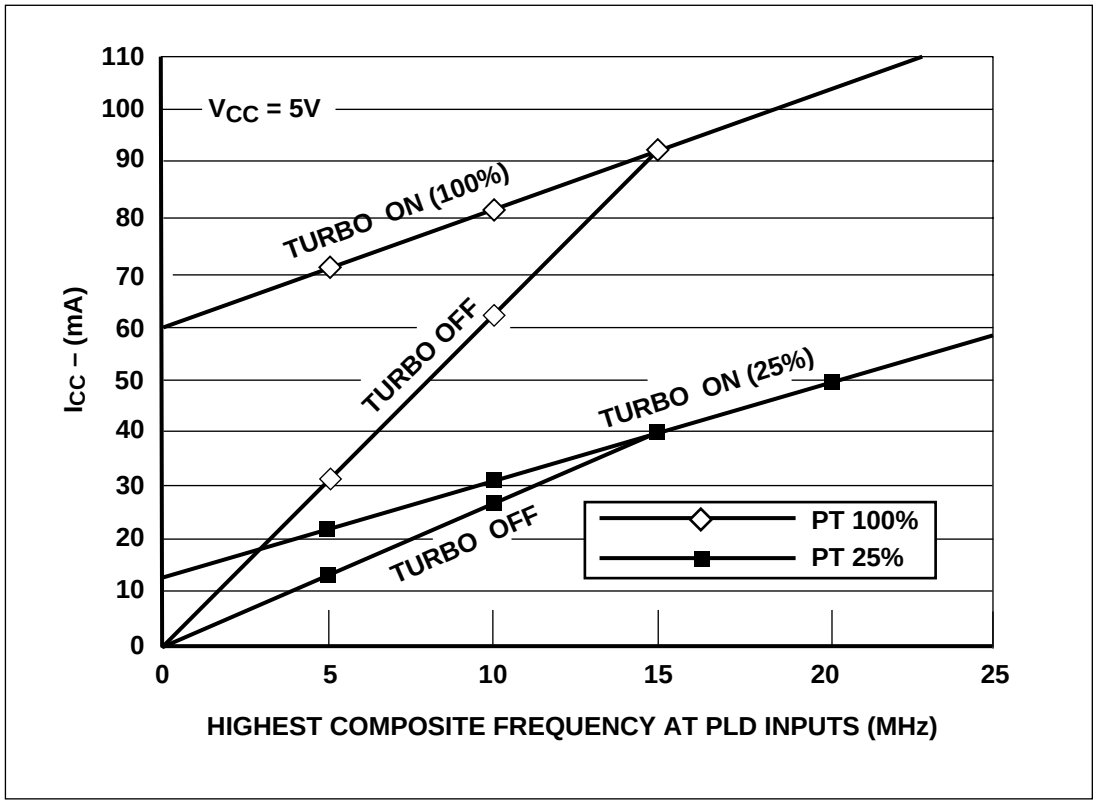
The following tables describe the AD/DC parameters of the PSD9XX family:

- ❏ DC Electrical Specification
- ❏ AC Timing Specification
  - PLD Timing
    - – Combinatorial Timing
  - Microcontroller Timing
    - – Read Timing
    - – Write Timing
    - – Power Down and Reset Timing

Following are issues concerning the parameters presented:

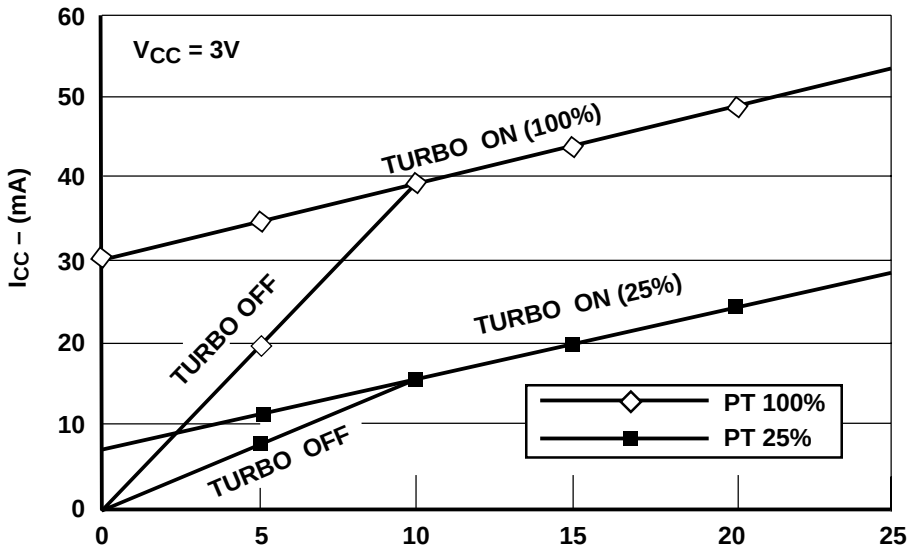
- ❏ In the DC specification the supply current is given for different modes of operation. Before calculating the total power consumption, determine the percentage of time that the PSD9XX is in each mode. Also, the supply power is considerably different if the Turbo bit is "OFF".
- ❏ The AC power component gives the PLD, Flash memory, and SRAM mA/MHz specification. Figure 26 shows the PLD mA/MHz as a function of the number of Product Terms (PT) used.
- ❏ In the PLD timing parameters, add the required delay when Turbo bit is "OFF".

Figure 26. PLD $I_{CC}$/FrequencyConsumption ($V_{CC}$ = 5 V ± 10%)



AC/DC Parameters (cont.)

Figure 26a. PLD $I_{CC}$/Frequency Consumption (PSD9XXFV Versions, $V_{CC}$ = 3 V)




Example of PSD9XX Typical Power Calculation at $V_{CC}$ = 5.0 V

| Conditions | | |
|---|---|---|
| Highest Composite PLD input frequency (Freq PLD) | = | 8 MHz |
| MCU ALE frequency (Freq ALE) | = | 4 MHz |
| % Flash Access | = | 80% |
| % SRAM access | = | 15% |
| % I/O access | = | 5% (no additional power above base) |
| Operational Modes | | |
| % Normal | = | 10% |
| % Power Down Mode | = | 90% |
| Number of product terms used (from fitter report) | = | 45 PT |
| % of total product terms | = | 45/153 = 29.4% |
| Turbo Mode | = | ON |

**Calculation (typical numbers used)**

$I_{CC}$ total = Ipwrdown x %pwrdown + %normal x ($I_{CC}$ (ac) + $I_{CC}$ (dc))
= Ipwrdown x %pwrdown + % normal x (%flash x 2.5 mA/MHz x Freq ALE
+ %SRAM x 1.5 mA/MHz x Freq ALE
+ % PLD x 2 mA/MHz x Freq PLD
+ #PT x 400 µA/PT
= 50 µA x 0.90 + 0.1 x (0.8 x 2.5 mA/MHz x 4 MHz
+ 0.15 x 1.5 mA/MHz x 4 MHz
+2 mA/MHz x 8 MHz
+ 45 x 0.4 mA/PT)
= 45 µA + 0.1 x (8 + 0.9 + 16 + 18 mA)
= 45 µA + 0.1 x 42.9
= 45 µA + 4.29 mA
**= 4.34 mA**

This is the operating power with no Flash writes or erases. Calculation is based on $I_{OUT}$ = 0 mA.

## AC/DC Parameters (cont.)

### Example of Typical Power Calculation at $V_{CC}$ = 5.0 V in Turbo Off Mode

| Conditions | | |
|---|---|---|
| Highest Composite PLD input frequency (Freq PLD) | = | 8 MHz |
| MCU ALE frequency (Freq ALE) | = | 4 MHz |
| % Flash Access | = | 80% |
| % SRAM access | = | 15% |
| % I/O access | = | 5% (no additional power above base) |
| Operational Modes | | |
| % Normal | = | 10% |
| % Power Down Mode | = | 90% |
| Number of product terms used (from fitter report) | = | 45 PT |
| % of total product terms | = | 45/153 = 29.4% |
| Turbo Mode | = | Off |

**Calculation (typical numbers used)**

$I_{CC}$ total = Ipwrdown x %pwrdown + %normal x ($I_{CC}$ (ac) + $I_{CC}$ (dc))
= Ipwrdown x %pwrdown + % normal x (%flash x 2.5 mA/MHz x Freq ALE
+ %SRAM x 1.5 mA/MHz x Freq ALE
+ % PLD x (from graph using Freq PLD))
= 50 µA x 0.90 + 0.1 x (0.8 x 2.5 mA/MHz x 4 MHz
+ 0.15 x 1.5 mA/MHz x 4 MHz
+ 24 mA)
= 45 µA + 0.1 x (8 + 0.9 + 24)
= 45 µA + 0.1 x 32.9
= 45 µA + 3.29 mA
**= 3.34 mA**

This is the operating power with no Flash writes or erases. Calculation is based on $I_{OUT}$ = 0 mA.

## PSD9XX DC Characteristics (5 V ± 10% Versions)

| Symbol | Parameter | | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|---|
| $V_{CC}$ | Supply Voltage | | All Speeds | 4.5 | 5 | 5.5 | V |
| $V_{IH}$ | High Level Input Voltage | | 4.5 V < $V_{CC}$ < 5.5 V | 2 | | $V_{CC}$ +.5 | V |
| $V_{IL}$ | Low Level Input Voltage | | 4.5 V < $V_{CC}$ < 5.5 V | –.5 | | 0.8 | V |
| $V_{IH1}$ | Reset High Level Input Voltage | | (Note 1) | .8 $V_{CC}$ | | $V_{CC}$ +.5 | V |
| $V_{IL1}$ | Reset Low Level Input Voltage | | (Note 1) | –.5 | | .2 $V_{CC}$ –.1 | V |
| $V_{HYS}$ | Reset Pin Hysteresis | | | 0.3 | | | V |
| $V_{LKO}$ | $V_{CC}$ Min for Flash Erase and Program | | | 2.5 | | 4.2 | V |
| $V_{OL}$ | Output Low Voltage | | $I_{OL}$ = 20 µA, $V_{CC}$ = 4.5 V | | 0.01 | 0.1 | V |
| | | | $I_{OL}$ = 8 mA, $V_{CC}$ = 4.5 V | | 0.25 | 0.45 | V |
| $V_{OH}$ | Output High Voltage Except $V_{STBY}$ On | | $I_{OH}$ = –20 µA, $V_{CC}$ = 4.5 V | 4.4 | 4.49 | | V |
| | | | $I_{OH}$ = –2 mA, $V_{CC}$ = 4.5 V | 2.4 | 3.9 | | V |
| $V_{OH_1}$ | Output High Voltage $V_{STBY}$ On | | $I_{OH_1}$ = 1 µA | $V_{SBY}$ – 0.8 | | | V |
| $V_{SBY}$ | SRAM Standby Voltage | | | 2.0 | | $V_{CC}$ | V |
| $I_{SBY}$ | SRAM Standby Current ($V_{STBY}$ Pin) | | $V_{CC}$ = 0 V | | 0.5 | 1 | µA |
| $I_{IDLE}$ | Idle Current ($V_{STBY}$ Pin) | | $V_{CC}$ > $V_{SBY}$ | –0.1 | | 0.1 | µA |
| $V_{DF}$ | SRAM Data Retention Voltage | | Only on $V_{STBY}$ | 2 | | | V |
| $I_{SB}$ | Standby Supply Current for Power Down Mode | | CSI > $V_{CC}$ –0.3 V (Notes 2 and 3) | | 75 | 200 | µA |
| $I_{LI}$ | Input Leakage Current | | $V_{SS}$ < $V_{IN}$ < $V_{CC}$ | –1 | ±.1 | 1 | µA |
| $I_{LO}$ | Output Leakage Current | | 0.45 < $V_{IN}$ < $V_{CC}$ | –10 | ±5 | 10 | µA |
| $I_{CC}$ (DC) (Note 5) | Operating Supply Current | PLD | PLD_TURBO = OFF, f = 0 MHz (Note 5) | | 0 | | mA |
| | | | PLD_TURBO = ON, f = 0 MHz | | 400 | 700 | µA/PT |
| | | Flash | During Flash Write/Erase Only | | 15 | 30 | mA |
| | | | Read Only, f = 0 MHz | | 0 | 0 | mA |
| | | SRAM | f = 0 MHz | | 0 | 0 | mA |
| $I_{CC}$ (AC) (Note 5) | PLD AC Adder | | | Fig. 26 (Note 4) | | | |
| | FLASH AC Adder | | | | 2.5 | 3.5 | mA/MHz |
| | SRAM AC Adder | | | | 1.5 | 3.0 | mA/MHz |

**NOTE:** 1. Reset input has hysteresis. $V_{IL1}$ is valid at or below .2$V_{CC}$ –.1. $V_{IH1}$ is valid at or above .8$V_{CC}$.
2. CSI deselected or internal Power Down mode is active.
3. PLD is in non-turbo mode and none of the inputs are switching
4. Refer to Figure 32 for PLD current calculation.
5. $I_{OUT}$ = 0 mA

## Microcontroller Interface – AC/DC Parameters

(5V ± 10% Versions)

AC Symbols for PLD Timing.

Example: $t_{AVLX}$ – Time from Address Valid to ALE Invalid.

### Signal Letters

- **A** – Address Input
- **C** – CEout Output
- **D** – Input Data
- **E** – E Input
- **L** – ALE Input
- **N** – Reset Input or Output
- **P** – Port Signal Output
- **Q** – Output Data
- **R** – $\overline{WR}$, $\overline{UDS}$, $\overline{LDS}$, $\overline{DS}$, IORD, $\overline{PSEN}$ Inputs
- **S** – Chip Select Input
- **T** – R/$\overline{W}$ Input
- **W** – Internal PDN Signal
- **B** – Vstby Output

### Signal Behavior

- **t** – Time
- **L** – Logic Level Low or ALE
- **H** – Logic Level High
- **V** – Valid
- **X** – No Longer a Valid Logic Level
- **Z** – Float
- **PW** – Pulse Width

# Microcontroller Interface – PSD9XX AC/DC Parameters

(5V ± 10% Versions)

## Read Timing (5 V ± 10% Versions)

| Symbol | Parameter | Conditions | -70 | | -90 | | -15 | | Turbo Off | Unit |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Min | Max | Min | Max | Min | Max | | |
| $t_{LVLX}$ | ALE or AS Pulse Width | | 15 | | 20 | | 28 | | | ns |
| $t_{AVLX}$ | Address Setup Time | (Note 3) | 4 | | 6 | | 10 | | | ns |
| $t_{LXAX}$ | Address Hold Time | (Note 3) | 7 | | 8 | | 11 | | | ns |
| $t_{AVQV}$ | Address Valid to Data Valid | (Note 3) | | 70 | | 90 | | 150 | Add 10 | ns |
| $t_{SLQV}$ | CS Valid to Data Valid | | | 75 | | 100 | | 150 | | ns |
| $t_{RLQV}$ | $\overline{RD}$ to Data Valid 8-Bit Bus | (Note 5) | | 24 | | 32 | | 40 | | ns |
| | $\overline{RD}$ or $\overline{PSEN}$ to Data Valid 8-Bit Bus, 8031, 80251 | (Note 2) | | 31 | | 38 | | 45 | | ns |
| $t_{RHQX}$ | $\overline{RD}$ Data Hold Time | (Note 1) | 0 | | 0 | | 0 | | | ns |
| $t_{RLRH}$ | $\overline{RD}$ Pulse Width | (Note 1) | 27 | | 32 | | 38 | | | ns |
| $t_{RHQZ}$ | $\overline{RD}$ to Data High-Z | (Note 1) | | 20 | | 25 | | 30 | | ns |
| $t_{EHEL}$ | E Pulse Width | | 27 | | 32 | | 38 | | | ns |
| $t_{THEH}$ | R/$\overline{W}$ Setup Time to Enable | | 6 | | 10 | | 18 | | | ns |
| $t_{ELTL}$ | R/$\overline{W}$ Hold Time After Enable | | 0 | | 0 | | 0 | | | ns |
| $t_{AVPV}$ | Address Input Valid to Address Output Delay | (Note 4) | | 20 | | 25 | | 30 | | ns |

**NOTES:**
1. $\overline{RD}$ timing has the same timing as $\overline{DS}$, $\overline{LDS}$, $\overline{UDS}$, and $\overline{PSEN}$ signals.
2. $\overline{RD}$ and $\overline{PSEN}$ have the same timing.
3. Any input used to select an internal PSD9XX function.
4. In multiplexed mode, latched addresses generated from ADIO delay to address output on any Port.
5. $\overline{RD}$ timing has the same timing as $\overline{DS}$, $\overline{LDS}$, and $\overline{UDS}$ signals.

# Microcontroller Interface – PSD9XX AC/DC Parameters

(5V ± 10% Versions)

## Write Timing (5 V ± 10% Versions)

| Symbol | Parameter | Conditions | -70 | | -90 | | -15 | | Unit |
|---|---|---|---|---|---|---|---|---|---|
| | | | Min | Max | Min | Max | Min | Max | |
| $t_{LVLX}$ | ALE or AS Pulse Width | | 15 | | 20 | | 28 | | |
| $t_{AVLX}$ | Address Setup Time | (Note 1) | 4 | | 6 | | 10 | | ns |
| $t_{LXAX}$ | Address Hold Time | (Note 1) | 7 | | 8 | | 11 | | ns |
| $t_{AVWL}$ | Address Valid to Leading Edge of $\overline{WR}$ | (Notes 1 and 3) | 8 | | 15 | | 20 | | ns |
| $t_{SLWL}$ | $\overline{CS}$ Valid to Leading Edge of $\overline{WR}$ | (Note 3) | 12 | | 15 | | 20 | | ns |
| $t_{DVWH}$ | $\overline{WR}$ Data Setup Time | (Note 3) | 25 | | 35 | | 45 | | ns |
| $t_{WHDX}$ | $\overline{WR}$ Data Hold Time | (Note 3) | 4 | | 5 | | 5 | | ns |
| $t_{WLWH}$ | $\overline{WR}$ Pulse Width | (Note 3) | 31 | | 35 | | 45 | | ns |
| $t_{WHAX1}$ | Trailing Edge of $\overline{WR}$ to Address Invalid | (Note 3) | 6 | | 8 | | 10 | | ns |
| $t_{WHAX2}$ | Trailing Edge of $\overline{WR}$ to DPLD Address Input Invalid | (Note 3 and 4) | 0 | | 0 | | 0 | | ns |
| $t_{WHPV}$ | Trailing Edge of $\overline{WR}$ to Port Output Valid Using I/O Port Data Register | (Note 3) | | 27 | | 30 | | 38 | ns |
| $t_{AVPV}$ | Address Input Valid to Address Output Delay | (Note 2) | | 20 | | 25 | | 30 | ns |

**NOTES:** 1. Any input used to select an internal PSD9XX function.
2. In multiplexed mode, latched addresses generated from ADIO delay to address output on any Port.
3. $\overline{WR}$ timing has the same timing as E, $\overline{LDS}$, $\overline{UDS}$, $\overline{WRL}$, and $\overline{WRH}$ signals.
4. Address Hold Time for DPLD inputs that are used to generate chip selects for internal PSD memory.

## PLD Combinatorial Timing (5 V ± 10%)

| Symbol | Parameter | Conditions | -70 | | -90 | | -15 | | Fast PT Aloc | TURBO OFF | Slew Rate (Note 1) | Unit |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Min | Max | Min | Max | Min | Max | | | | |
| $t_{PD}$ | PLD Input Pin/Feedback to PLD Combinatorial Output | | | 20 | | 25 | | 32 | Add 2 | Add 10 | Sub 2 | ns |
| $t_{ARD}$ | PLD Array Delay | Any Micro⇔Cell | | 11 | | 16 | | 22 | Add 2 | | | ns |

**NOTE:** 1. Fast Slew Rate output available on PA[3:0], PB[3:0], and PD[2:0].

# Microcontroller Interface – PSD9XX AC/DC Parameters

(5V ± 10% Versions)

## Power Down Timing (5 V ± 10%)

| Symbol | Parameter | Conditions | -70 | | -90 | | -15 | | Unit |
|---|---|---|---|---|---|---|---|---|---|
| | | | Min | Max | Min | Max | Min | Max | |
| $t_{LVDV}$ | ALE Access Time from Power Down | | | 80 | | 90 | | 150 | ns |
| $t_{CLWH}$ | Maximum Delay from APD Enable to Internal PDN Valid Signal | Using CLKIN Input | | | 15 * $t_{CLCL}$ (µs) (Note 1) | | | | µs |

**NOTE:** 1. $t_{CLCL}$ is the CLKIN clock period.

## $V_{stbyon}$ Timing (5 V ± 10%)

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $t_{BVBH}$ | Vstby Detection to Vstbyon Output High | (Note 1) | | 20 | | µs |
| $t_{BXBL}$ | $V_{stby}$ Off Detection to $V_{stbyon}$ Output Low | (Note 1) | | 20 | | µs |

**NOTE:** 1. Vstbyon is measured at $V_{CC}$ ramp rate of 2 ms.

## Reset Pin Timing (5 V ± 10%)

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $t_{NLNH}$ | Warm RESET Active Low Time (Note 1) | | 150 | | | ns |
| $t_{OPR}$ | RESET High to Operational Device | | | | 120 | ns |
| $t_{NLNH-PO}$ | Power On Reset Active Low Time | | 1 | | | ms |
| $t_{NLNH-A}$ | Warm Reset, will abort and reset Flash programming/erase cycles to Read mode. (Note 2) | | 25 | | | µs |

**NOTE:** 1. RESET will not reset Flash programming/erase cycles.
2. RESET will abort Flash programming or erase cycle. For PSD934F2 and PSD954F2 only.

## Microcontroller Interface – PSD9XX AC/DC Parameters

(5V ±10% Versions)

### Flash Program, Write and Erase Times (5 V ± 10%)

| Symbol | Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| | Flash Bulk Erase (Preprogrammed to 00) (Note 1) | | 3 | 30 | sec |
| | Flash Bulk Erase (Not Preprogrammed) | | 5 | | sec |
| $t_{WHQV3}$ | Sector Erase (Preprogrammed to 00) | | 1 | 30 | sec |
| $t_{WHQV2}$ | Sector Erase (Not Preprogrammed) | | 2.2 | | sec |
| $t_{WHQV1}$ | Byte Program | | 14 | 1200 | µs |
| | Program/Erase Cycles (Per Sector) | 100,000 | | | cycles |
| $t_{WHWLO}$ | Sector Erase Time-Out | | 100 | | µs |
| $t_{Q7VQV}$ | DQ7 Valid to Output (DQ7-0) Valid (Data Polling) (Note 2) | | | 30 | ns |

**NOTE:** 1. Programmed to all zeros before erase.
2. The polling status DQ7 is valid tQ7VQV ns before the data byte DQ0-7 is valid for reading.

### ISC Timing (5 V ± 10%)

| Symbol | Parameter | Conditions | -70 | | -90 | | -15 | | Unit |
|---|---|---|---|---|---|---|---|---|---|
| | | | Min | Max | Min | Max | Min | Max | |
| $t_{ISCCF}$ | TCK Clock Frequency (except for PLD) | (Note 1) | | 20 | | 18 | | 14 | MHz |
| $t_{ISCCH}$ | TCK Clock High Time | (Note 1) | 23 | | 26 | | 31 | | ns |
| $t_{ISCCL}$ | TCK Clock Low Time | (Note 1) | 23 | | 26 | | 31 | | ns |
| $t_{ISCCF-P}$ | TCK Clock Frequency (for PLD only) | (Note 2) | | 2 | | 2 | | 2 | MHz |
| $t_{ISCCH-P}$ | TCK Clock High Time(for PLD only) | (Note 2) | 240 | | 240 | | 240 | | ns |
| $t_{ISCCL-P}$ | TCK Clock Low Time(for PLD only) | (Note 2) | 240 | | 240 | | 240 | | ns |
| $t_{ISCPSU}$ | ISC Port Set Up Time | | 7 | | 8 | | 10 | | ns |
| $t_{ISCPH}$ | ISC Port Hold Up Time | | 5 | | 5 | | 5 | | ns |
| $t_{ISCPCO}$ | ISC Port Clock to Output | | | 21 | | 23 | | 25 | ns |
| $t_{ISCPZV}$ | ISC Port High-Impedance to Valid Output | | | 21 | | 23 | | 25 | ns |
| $t_{ISCPVZ}$ | ISC Port Valid Output to High-Impedance | | | 21 | | 23 | | 25 | ns |

**NOTES:** 1. For “non-PLD” programming, erase or in ISC by-pass mode.
2. For program or erase PLD only.

## PSD9XXFV DC Characteristics (3.0 V to 3.6 V Versions) Advance Information

| Symbol | Parameter | | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|---|
| $V_{CC}$ | Supply Voltage | | All Speeds | 3.0 | | 3.6 | V |
| $V_{IH}$ | High Level Input Voltage | | 3.0 V < $V_{CC}$ < 3.6 V | .7 $V_{CC}$ | | $V_{CC}$ +.5 | V |
| $V_{IL}$ | Low Level Input Voltage | | 3.0 V < $V_{CC}$ < 3.6 V | –.5 | | 0.8 | V |
| $V_{IH1}$ | Reset High Level Input Voltage | | (Note 1) | .8 $V_{CC}$ | | $V_{CC}$ +.5 | V |
| $V_{IL1}$ | Reset Low Level Input Voltage | | (Note 1) | –.5 | | .2 $V_{CC}$ –.1 | V |
| $V_{HYS}$ | Reset Pin Hysteresis | | | 0.3 | | | V |
| $V_{LKO}$ | $V_{CC}$ Min for Flash Erase and Program | | | 1.5 | | 2.2 | V |
| $V_{OL}$ | Output Low Voltage | | $I_{OL}$ = 20 µA, $V_{CC}$ = 3.0 V | | 0.01 | 0.1 | V |
| | | | $I_{OL}$ = 4 mA, $V_{CC}$ = 3.0 V | | 0.15 | 0.45 | V |
| $V_{OH}$ | Output High Voltage Except $V_{STBY}$ On | | $I_{OH}$ = –20 µA, $V_{CC}$ = 3.0 V | 2.9 | 2.99 | | V |
| | | | $I_{OH}$ = –1 mA, $V_{CC}$ = 3.0 V | 2.7 | 2.8 | | V |
| $V_{OH_1}$ | Output High Voltage $V_{STBY}$ On | | $I_{OH_1}$ = –1 µA | $V_{SBY}$ – 0.8 | | | V |
| $V_{SBY}$ | SRAM Standby Voltage | | | 2.0 | | $V_{CC}$ | V |
| $I_{SBY}$ | SRAM Standby Current ($V_{STBY}$ Pin) | | $V_{CC}$ = 0 V | | 0.5 | 1 | µA |
| $I_{IDLE}$ | Idle Current ($V_{STBY}$ Pin) | | $V_{CC}$ > $V_{SBY}$ | –0.1 | | 0.1 | µA |
| $V_{DF}$ | SRAM Data Retention Voltage | | Only on $V_{STBY}$ | 2 | | | V |
| $I_{SB}$ | Standby Supply Current for Power Down Mode | | $\overline{CSI}$ >$V_{CC}$ –0.3 V (Notes 2 and 3) | | 25 | 100 | µA |
| $I_{LI}$ | Input Leakage Current | | $V_{SS}$ < $V_{IN}$ < $V_{CC}$ | –1 | ±.1 | 1 | µA |
| $I_{LO}$ | Output Leakage Current | | 0.45 < $V_{IN}$ < $V_{CC}$ | –10 | ±5 | 10 | µA |
| $I_{CC}$ (DC) (Note 5) | Operating Supply Current | PLD Only | PLD_TURBO = OFF, f = 0 MHz (Note 3) | | 0 | | mA |
| | | | PLD_TURBO = ON, f = 0 MHz | | 200 | 400 | µA/PT |
| | | FLASH | During FLASH or Write/Erase Only | | 10 | 25 | mA |
| | | | Read Only, f = 0 MHz | | 0 | 0 | mA |
| | | SRAM | f = 0 MHz | | 0 | 0 | mA |
| $I_{CC}$ (AC) (Note 5) | PLD AC Adder | | | | (Note 4) | | Figure26a |
| | FLASH AC Adder | | | | 1.5 | 2.0 | mA/MHz |
| | SRAM AC Adder | | | | 0.8 | 1.5 | mA/MHz |

**NOTES:**
1. Reset input has hysteresis. $V_{IL1}$ is valid at or below .2$V_{CC}$ –.1. $V_{IH1}$ is valid at or above .8$V_{CC}$.
2. $\overline{CSI}$ deselected or internal PD is active.
3. PLD is in non-turbo mode and none of the inputs are switching.
4. Refer to Figure 26a for PLD current calculation.
5. $I_{OUT}$ = 0 mA.

## Microcontroller Interface – PSD9XXFV AC/DC Parameters

(3 V Versions)

### Read Timing (3 V Versions)

| Symbol | Parameter | Conditions | -12 | | -15 | | -20 | | Turbo Off | Unit |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Min | Max | Min | Max | Min | Max | | |
| $t_{LVLX}$ | ALE or AS Pulse Width | | 26 | | 26 | | 30 | | | ns |
| $t_{AVLX}$ | Address Setup Time | (Note 3) | 9 | | 10 | | 12 | | | ns |
| $t_{LXAX}$ | Address Hold Time | (Note 3) | 9 | | 12 | | 14 | | | ns |
| $t_{AVQV}$ | Address Valid to Data Valid | (Note 3) | | 120 | | 150 | | 200 | Add 20 | ns |
| $t_{SLQV}$ | $\overline{CS}$ Valid to Data Valid | | | 120 | | 150 | | 200 | | ns |
| $t_{RLQV}$ | $\overline{RD}$ to Data Valid 8-Bit Bus | (Note 5) | | 35 | | 35 | | 40 | | ns |
| | $\overline{RD}$ or $\overline{PSEN}$ to Data Valid 8-Bit Bus, 8031, 80251 | (Note 2) | | 45 | | 50 | | 55 | | ns |
| $t_{RHQX}$ | $\overline{RD}$ Data Hold Time | (Note 1) | 0 | | 0 | | 0 | | | ns |
| $t_{RLRH}$ | $\overline{RD}$ Pulse Width | (Note 1) | 38 | | 40 | | 45 | | | ns |
| $t_{RHQZ}$ | $\overline{RD}$ to Data High-Z | (Note 1) | | 38 | | 40 | | 45 | | ns |
| $t_{EHEL}$ | E Pulse Width | | 40 | | 45 | | 52 | | | ns |
| $t_{THEH}$ | R/$\overline{W}$ Setup Time to Enable | | 15 | | 18 | | 20 | | | ns |
| $t_{ELTL}$ | R/$\overline{W}$ Hold Time After Enable | | 0 | | 0 | | 0 | | | ns |
| $t_{AVPV}$ | Address Input Valid to Address Output Delay | (Note 4) | | 33 | | 35 | | 40 | | ns |

**NOTES:** 1. $\overline{RD}$ timing has the same timing as $\overline{DS}$, $\overline{LDS}$, $\overline{UDS}$, and $\overline{PSEN}$ signals.
2. $\overline{RD}$ and $\overline{PSEN}$ have the same timing for 8031.
3. Any input used to select an internal PSD813F function.
4. In multiplexed mode latched address generated from ADIO delay to address output on any Port.
5. RD timing has the same timing as $\overline{DS}$, $\overline{LDS}$, and $\overline{UDS}$ signals.

# Microcontroller Interface – PSD9XXFV AC/DC Parameters

(3 V Versions)

## Write Timing (3 V Versions)

| Symbol | Parameter | Conditions | -12 | | -15 | | -20 | | Unit |
|---|---|---|---|---|---|---|---|---|---|
| | | | Min | Max | Min | Max | Min | Max | |
| $t_{LVLX}$ | ALE or AS Pulse Width | | 26 | | 26 | | 30 | | |
| $t_{AVLX}$ | Address Setup Time | (Note 1) | 9 | | 10 | | 12 | | ns |
| $t_{LXAX}$ | Address Hold Time | (Note 1) | 9 | | 12 | | 14 | | ns |
| $t_{AVWL}$ | Address Valid to Leading Edge of $\overline{WR}$ | (Notes 1 and 3) | 17 | | 20 | | 25 | | ns |
| $t_{SLWL}$ | $\overline{CS}$ Valid to Leading Edge of $\overline{WR}$ | (Note 3) | 17 | | 20 | | 25 | | ns |
| $t_{DVWH}$ | $\overline{WR}$ Data Setup Time | (Note 3) | 45 | | 45 | | 50 | | ns |
| $t_{WHDX}$ | $\overline{WR}$ Data Hold Time | (Note 3) | 7 | | 8 | | 10 | | ns |
| $t_{WLWH}$ | $\overline{WR}$ Pulse Width | (Note 3) | 46 | | 48 | | 53 | | ns |
| $t_{WHAX1}$ | Trailing Edge of $\overline{WR}$ to Address Invalid | (Note 3) | 10 | | 12 | | 17 | | ns |
| $t_{WHAX2}$ | Trailing Edge of $\overline{WR}$ to DPLD Address Input Invalid | (Notes 3 and 4) | 0 | | 0 | | 0 | | ns |
| $t_{WHPV}$ | Trailing Edge of $\overline{WR}$ to Port Output Valid Using I/O Port Data Register | (Note 3) | | 33 | | 35 | | 40 | ns |
| $t_{AVPV}$ | Address Input Valid to Address Output Delay | (Note 2) | | 33 | | 35 | | 40 | ns |

**NOTES:** 1. Any input used to select an internal PSD813F function.
2. In multiplexed mode, latched addresses generated from ADIO delay to address output on any Port.
3. $\overline{WR}$ timing has the same timing as E, $\overline{LDS}$, $\overline{UDS}$, $\overline{WRL}$, and $\overline{WRH}$ signals.
4. Address hold time for DPLD inputs that are used to generate chip selects for internal PSD memory.

## PLD Combinatorial Timing (3 V Versions)

| Symbol | Parameter | Conditions | -12 | | -15 | | -20 | | Fast PT Aloc | TURBO OFF | Slew Rate (Note 1) | Unit |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Min | Max | Min | Max | Min | Max | | | | |
| $t_{PD}$ | PLD Input Pin/Feedback to PLD Combinatorial Output | | | 40 | | 45 | | 50 | Add 4 | Add 20 | Sub 6 | ns |
| $t_{ARD}$ | PLD Array Delay | Any Micro⇔Cell | | 25 | | 29 | | 33 | Add 4 | | | ns |

**NOTE:** 1. Fast Slew Rate output available on PA[3:0], PB[3:0], and PD[2:0].

# Microcontroller Interface – PSD9XXFV AC/DC Parameters

(3 V Versions)

## Power Down Timing (3 V Versions)

| Symbol | Parameter | Conditions | -12 | | -15 | | -20 | | Unit |
|---|---|---|---|---|---|---|---|---|---|
| | | | Min | Max | Min | Max | Min | Max | |
| $t_{LVDV}$ | ALE Access Time from Power Down | | | 145 | | 150 | | 200 | ns |
| $t_{CLWH}$ | Maximum Delay from APD Enable to Internal PDN Valid Signal | Using CLKIN Input | | | 15 * $t_{CLCL}$ (µs) (Note 1) | | | | µs |

**NOTE:** 1. $t_{CLCL}$ is the CLKIN clock period.

## $V_{stbyon}$ Timing (3 V Versions)

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $t_{BVBH}$ | $V_{stby}$ Detection to $V_{stbyon}$ Output High | (Note 1) | | 20 | | µs |
| $t_{BXBL}$ | $V_{stby}$ Off Detection to $V_{stbyon}$ Output Low | (Note 1) | | 20 | | µs |

**NOTE:** 1. Vstbyon is measured at $V_{CC}$ ramp rate of 2 ms.

## Reset Pin Timing (3 V Versions)

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $t_{NLNH}$ | Warm RESET Active Low Time (Note 1) | | 300 | | | ns |
| $t_{OPR}$ | RESET High to Operational Device | | | | 300 | ns |
| $t_{NLNH-PO}$ | Power On Reset Active Low Time | | 1 | | | ms |
| $t_{NLNH-A}$ | Warm Reset, will abort and reset Flash programming/erase cycles to Read mode. For PSD9X4FV only. | | 25 | | | µs |

**NOTE:** 1. RESET will not reset Flash programming/erase cycles.
2. RESET will abort Flash programming or erase cycle.

# Microcontroller Interface – PSD9XXFV AC/DC Parameters

(3 V Versions)

## Flash Program, Write and Erase Times (3 V Versions)

| Symbol | Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| | Flash Bulk Erase (Preprogrammed to 00) (Note 1) | | 3 | 30 | sec |
| | Flash Bulk Erase (Not Preprogrammed) | | 5 | | sec |
| $t_{WHQV3}$ | Sector Erase (Preprogrammed to 00) | | 1 | 30 | sec |
| $t_{WHQV2}$ | Sector Erase (Not Preprogrammed) | | 2.2 | | sec |
| $t_{WHQV1}$ | Byte Program | | 14 | 1200 | µs |
| | Program/Erase Cycles (Per Sector) | 100,000 | | | cycles |
| $t_{WHWLO}$ | Sector Erase Time-Out | | 100 | | µs |
| $t_{Q7VQV}$ | DQ7 Valid to Output (DQ7-0) Valid (Data Polling) (Note 2) | | | 30 | ns |

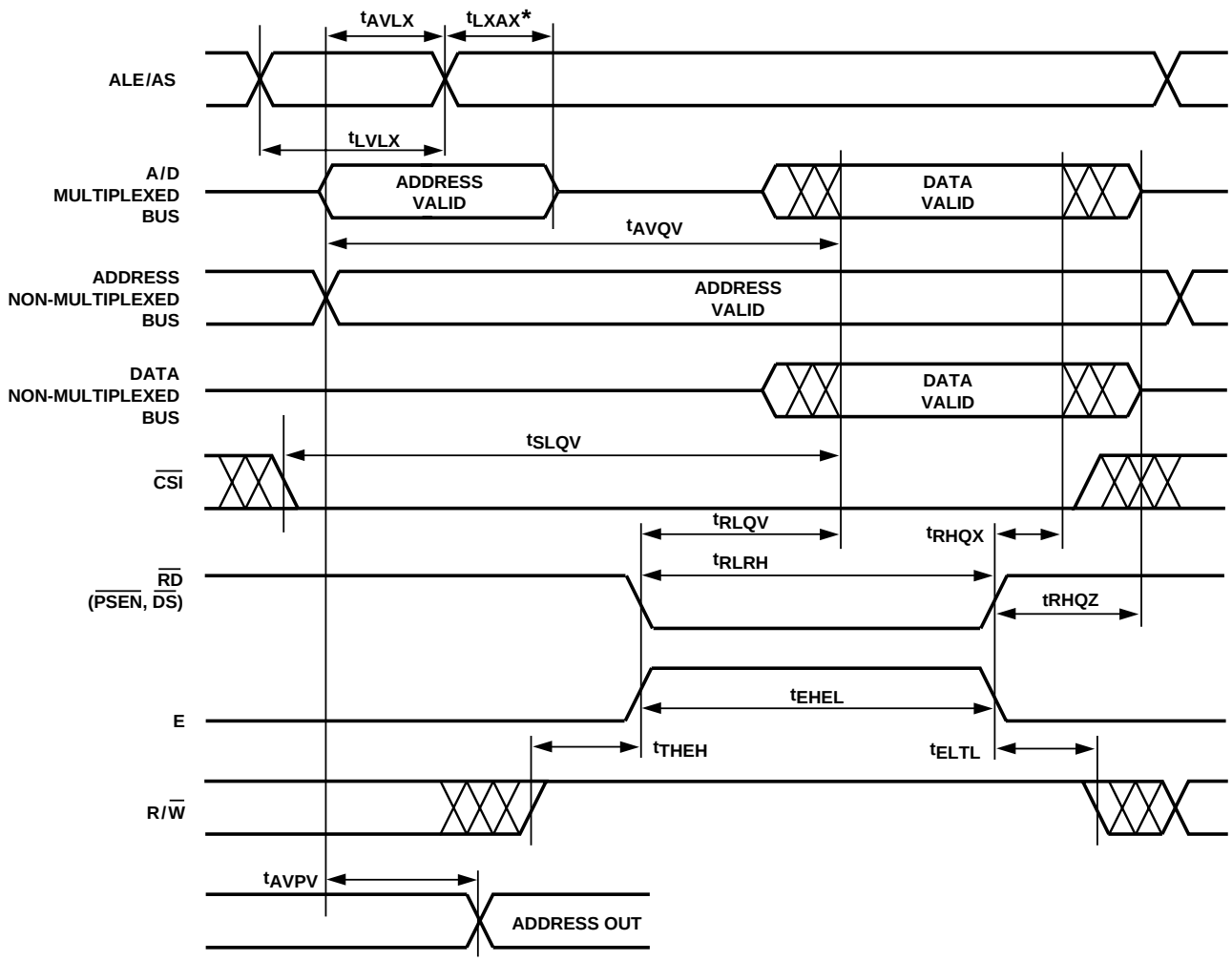
**NOTE:** 1. Programmed to all zeros before erase.
2. The polling status DQ7 is valid tQ7VQV ns before the data byte DQ0-7 is valid for reading.

## ISC Timing (3 V Versions)

| Symbol | Parameter | Conditions | -12 | | -15 | | -20 | | Unit |
|---|---|---|---|---|---|---|---|---|---|
| | | | Min | Max | Min | Max | Min | Max | |
| $t_{ISCCF}$ | TCK Clock Frequency (except for PLD) | (Note 1) | | 12 | | 10 | | 9 | MHz |
| $t_{ISCCH}$ | TCK Clock High Time | (Note 1) | 40 | | 45 | | 51 | | ns |
| $t_{ISCCL}$ | TCK Clock Low Time | (Note 1) | 40 | | 45 | | 51 | | ns |
| $t_{ISCCF-P}$ | TCK Clock Frequency (for PLD only) | (Note 2) | | 2 | | 2 | | 2 | MHz |
| $t_{ISCCH-P}$ | TCK Clock High Time (for PLD only) | (Note 2) | 240 | | 240 | | 240 | | ns |
| $t_{ISCCL-P}$ | TCK Clock Low Time (for PLD only) | (Note 2) | 240 | | 240 | | 240 | | ns |
| $t_{ISCPSU}$ | ISC Port Set Up Time | | 12 | | 13 | | 15 | | ns |
| $t_{ISCPH}$ | ISC Port Hold Up Time | | 5 | | 5 | | 5 | | ns |
| $t_{ISCPCO}$ | ISC Port Clock to Output | | | 30 | | 36 | | 40 | ns |
| $t_{ISCPZV}$ | ISC Port High-Impedance to Valid Output | | | 30 | | 36 | | 40 | ns |
| $t_{ISCPVZ}$ | ISC Port Valid Output to High-Impedance | | | 30 | | 36 | | 40 | ns |

**NOTES:** 1. For “non-PLD” programming, erase or in ISC by-pass mode.
2. For program or erase PLD only.

Figure 27. Read Timing




*$t_{AVLX}$ and $t_{LXAX}$ are not required for 80C251 in Page Mode or 80C51XA in Burst Mode.

Figure 28. Write Timing

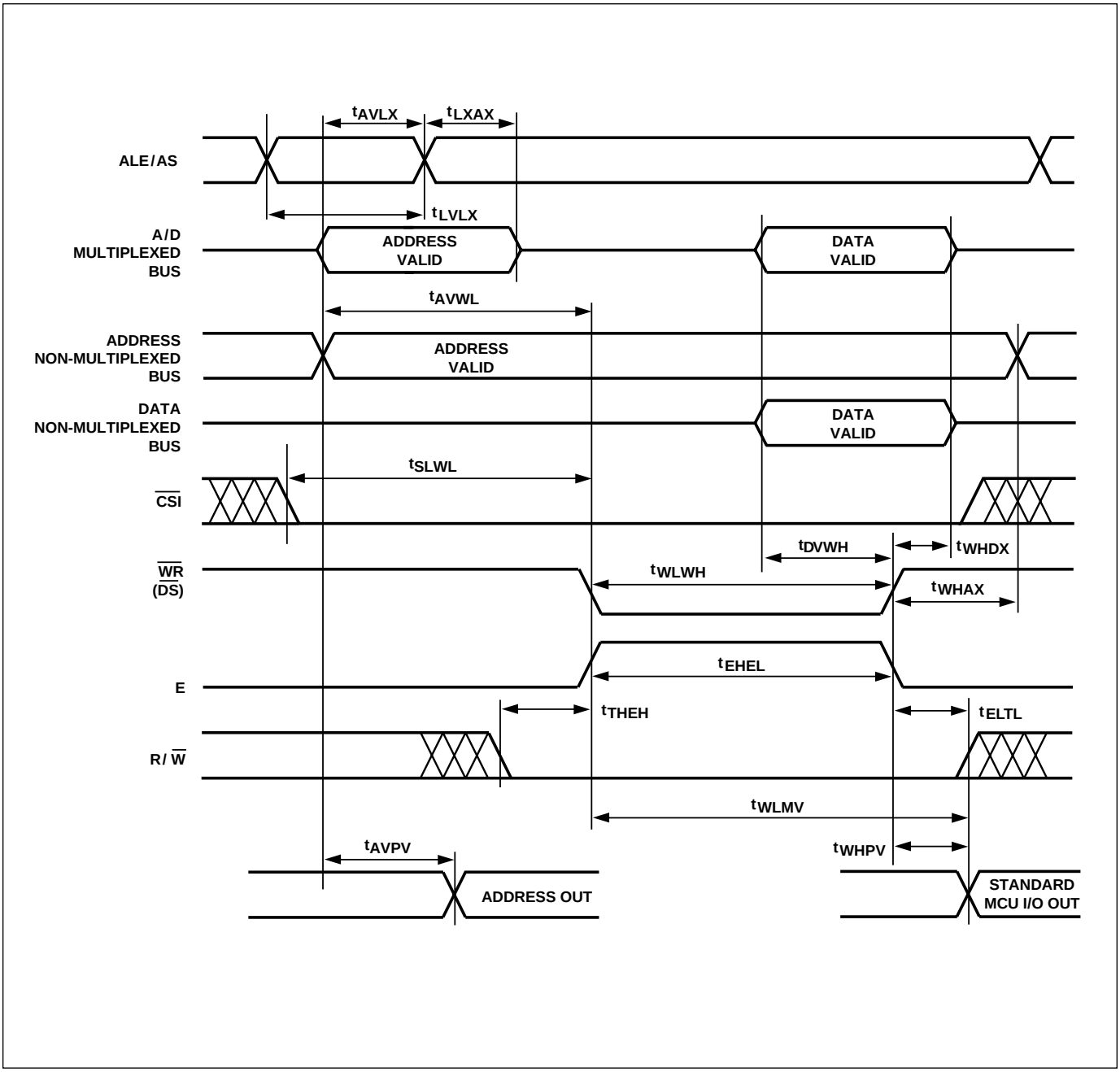


Figure 29. Combinatorial Timing – PLD

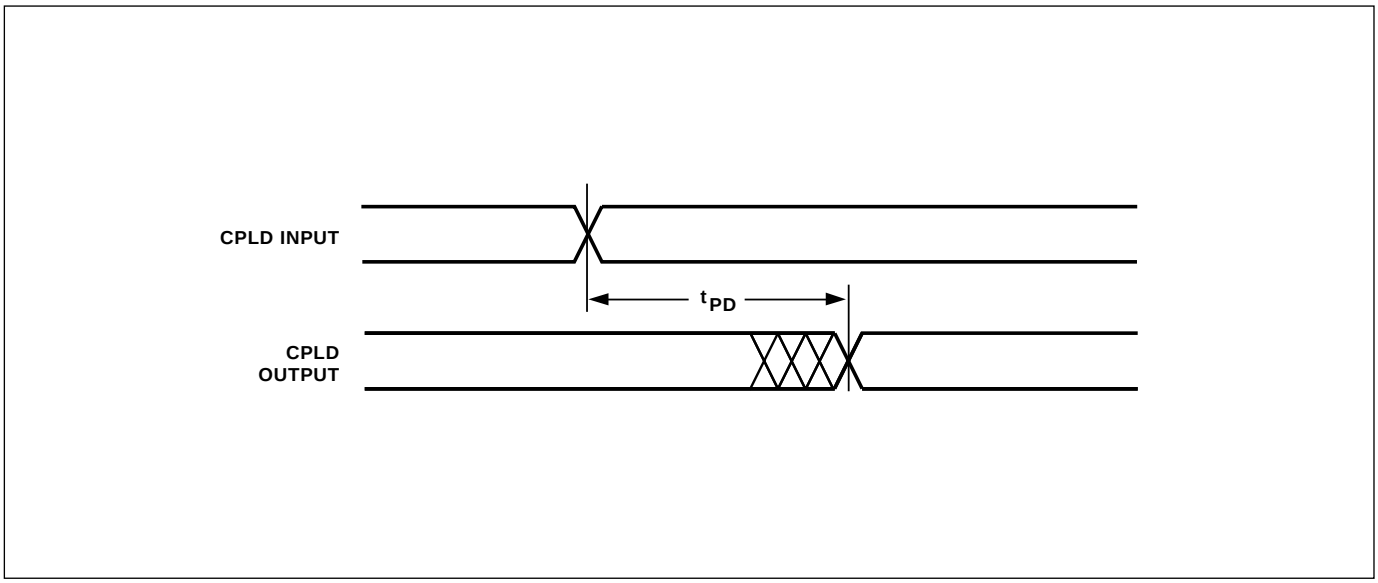



Figure 30. ISC Timing

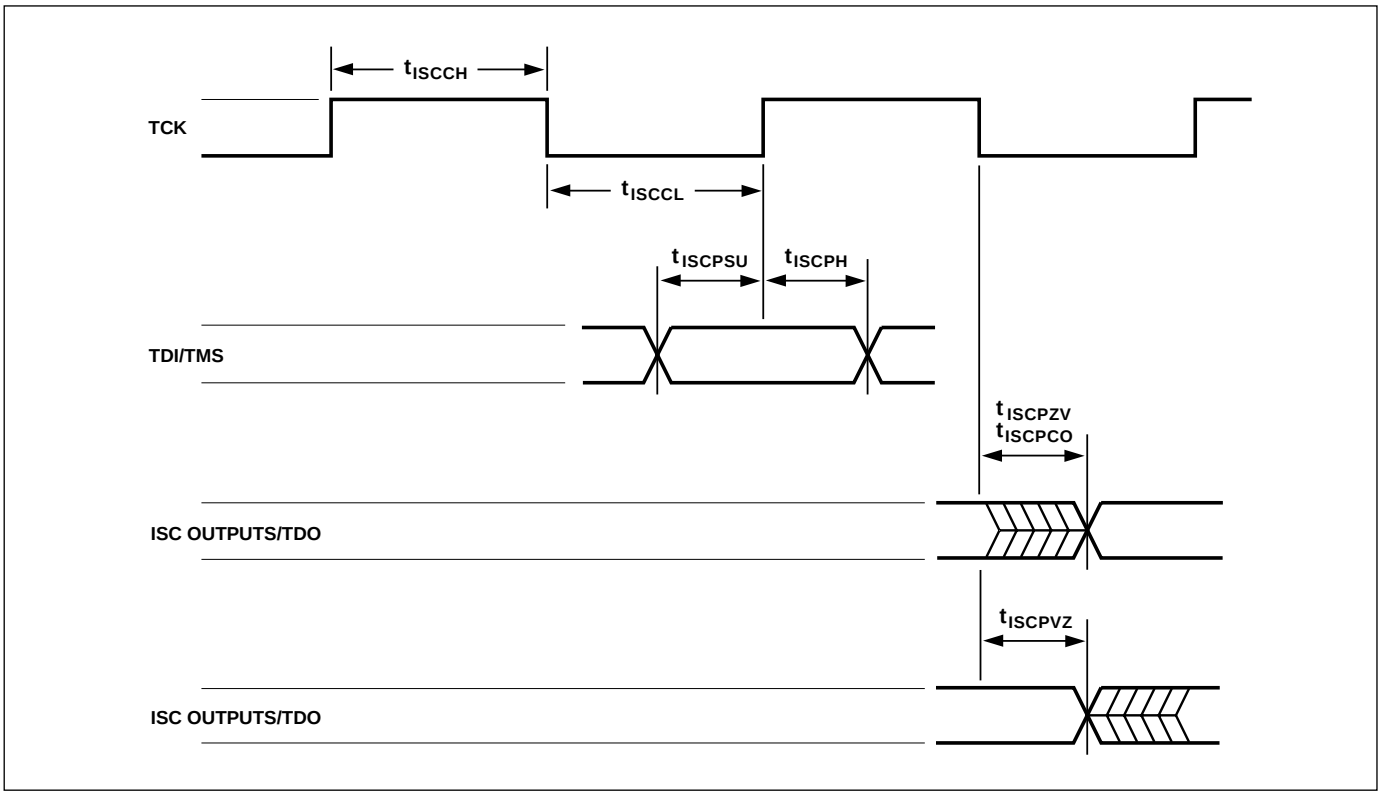


Figure 31. Reset Timing

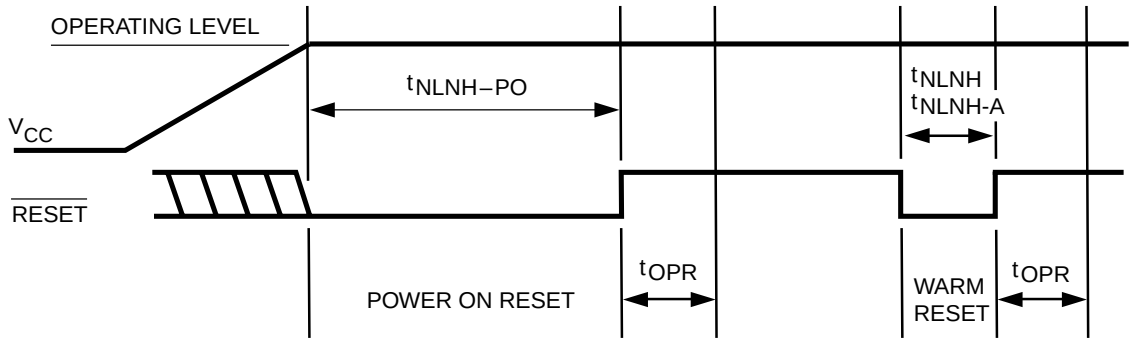
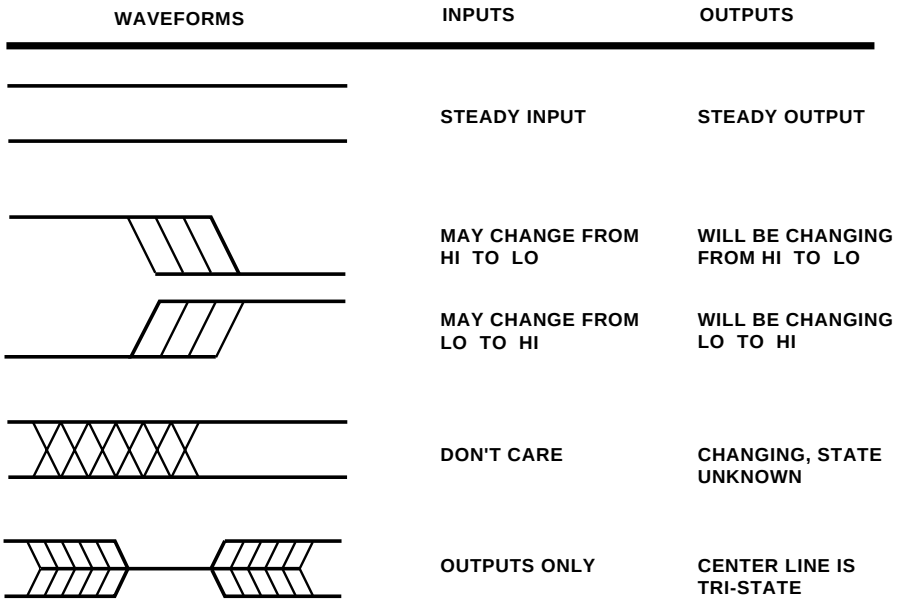



Figure 32. Key to Switching Waveforms



## Pin Capacitance

*$T_A$ = 25 °C, f = 1 MHz*

| Symbol | Parameter[1] | Conditions | Typical[2] | Max | Unit |
|---|---|---|---|---|---|
| $C_{IN}$ | Capacitance (for input pins only) | $V_{IN}$ = 0 V | 4 | 6 | pF |
| $C_{OUT}$ | Capacitance (for input/output pins) | $V_{OUT}$ = 0 V | 8 | 12 | pF |
| $C_{VPP}$ | Capacitance (for CNTL2/$V_{PP}$) | $V_{PP}$ = 0 V | 18 | 25 | pF |

**NOTES:** 1. These parameters are only sampled and are not 100% tested.
2. Typical values are for $T_A$ = 25°C and nominal supply voltages.

Figure 33. AC Testing Input/Output Waveform

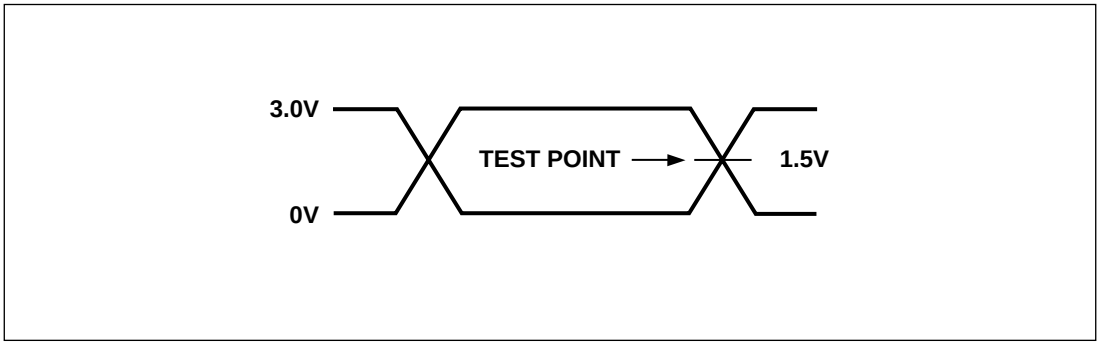
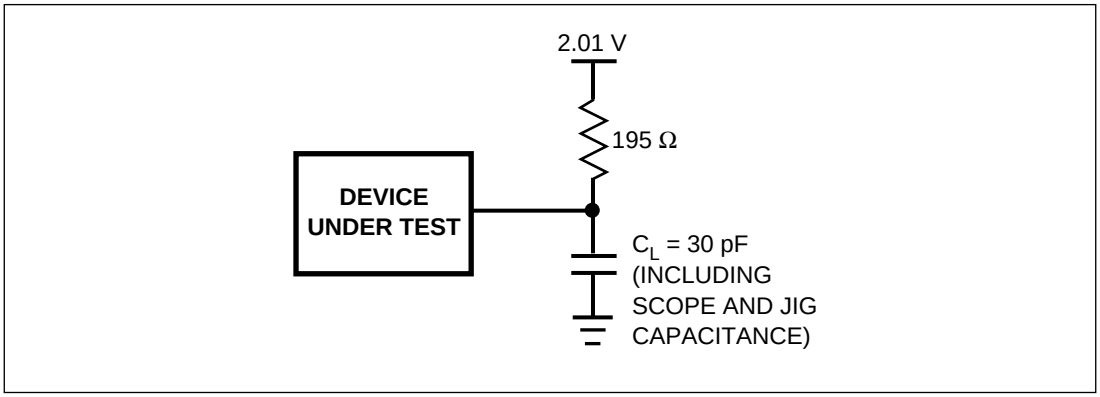



Figure 34. AC Testing Load Circuit




## Programming

Upon delivery from ST, the PSD9XX device has all bits in the PLDs and memories in the “1” or high state. The configuration bits are in the “0” or low state. The code, configuration, and PLDs logic are loaded through the procedure of programming.

Information for programming the device is available directly from ST. Please contact your local sales representative. (See the last page.)

## PSD9XX Pin Assignments

52-Pin Plastic Leaded Chip Carrier (PLCC) (Package Type J)

| Pin No. | Pin Assignments | Pin No. | Pin Assignments |
|---|---|---|---|
| 1 | GND | 27 | PA2 |
| 2 | PB5 | 28 | PA1 |
| 3 | PB4 | 29 | PA0 |
| 4 | PB3 | 30 | AD0 |
| 5 | PB2 | 31 | AD1 |
| 6 | PB1 | 32 | AD2 |
| 7 | PB0 | 33 | AD3 |
| 8 | PD2 | 34 | AD4 |
| 9 | PD1 | 35 | AD5 |
| 10 | PD0 | 36 | AD6 |
| 11 | PC7 | 37 | AD7 |
| 12 | PC6 | 38 | $V_{CC}$ |
| 13 | PC5 | 39 | AD8 |
| 14 | PC4 | 40 | AD9 |
| 15 | $V_{CC}$ | 41 | AD10 |
| 16 | GND | 42 | AD11 |
| 17 | PC3 | 43 | AD12 |
| 18 | PC2 (VSTBY) | 44 | AD13 |
| 19 | PC1 | 45 | AD14 |
| 20 | PC0 | 46 | AD15 |
| 21 | PA7 | 47 | CNTL0 |
| 22 | PA6 | 48 | $\overline{\text{RESET}}$ |
| 23 | PA5 | 49 | CNTL2 |
| 24 | PA4 | 50 | CNTL1 |
| 25 | PA3 | 51 | PB7 |
| 26 | GND | 52 | PB6 |

## PSD9XX Pin Assignments (cont.)

52-Pin Plastic Quad Flatpack (PQFP) (Package Type M)

| Pin No. | Pin Assignments | Pin No. | Pin Assignments |
|---|---|---|---|
| 1 | PD2 | 27 | AD4 |
| 2 | PD1 | 28 | AD5 |
| 3 | PD0 | 29 | AD6 |
| 4 | PC7 | 30 | AD7 |
| 5 | PC6 | 31 | VCC |
| 6 | PC5 | 32 | AD8 |
| 7 | PC4 | 33 | AD9 |
| 8 | VCC | 34 | AD10 |
| 9 | GND | 35 | AD11 |
| 10 | PC3 | 36 | AD12 |
| 11 | PC2 | 37 | AD13 |
| 12 | PC1 | 38 | AD14 |
| 13 | PC0 | 39 | AD15 |
| 14 | PA7 | 40 | CNTL0 |
| 15 | PA6 | 41 | $\overline{\text{RESET}}$ |
| 16 | PA5 | 42 | CNTL2 |
| 17 | PA4 | 43 | CNTL1 |
| 18 | PA3 | 44 | PB7 |
| 19 | GND | 45 | PB6 |
| 20 | PA2 | 46 | GND |
| 21 | PA1 | 47 | PB5 |
| 22 | PA0 | 48 | PB4 |
| 23 | AD0 | 49 | PB3 |
| 24 | AD1 | 50 | PB2 |
| 25 | AD2 | 51 | PB1 |
| 26 | AD3 | 52 | PB0 |

## PSD9XX Package Information

Figure 35. Drawing J7 – 52-Pin Plastic Leaded Chip Carrier (PLCC) (Package Type J)

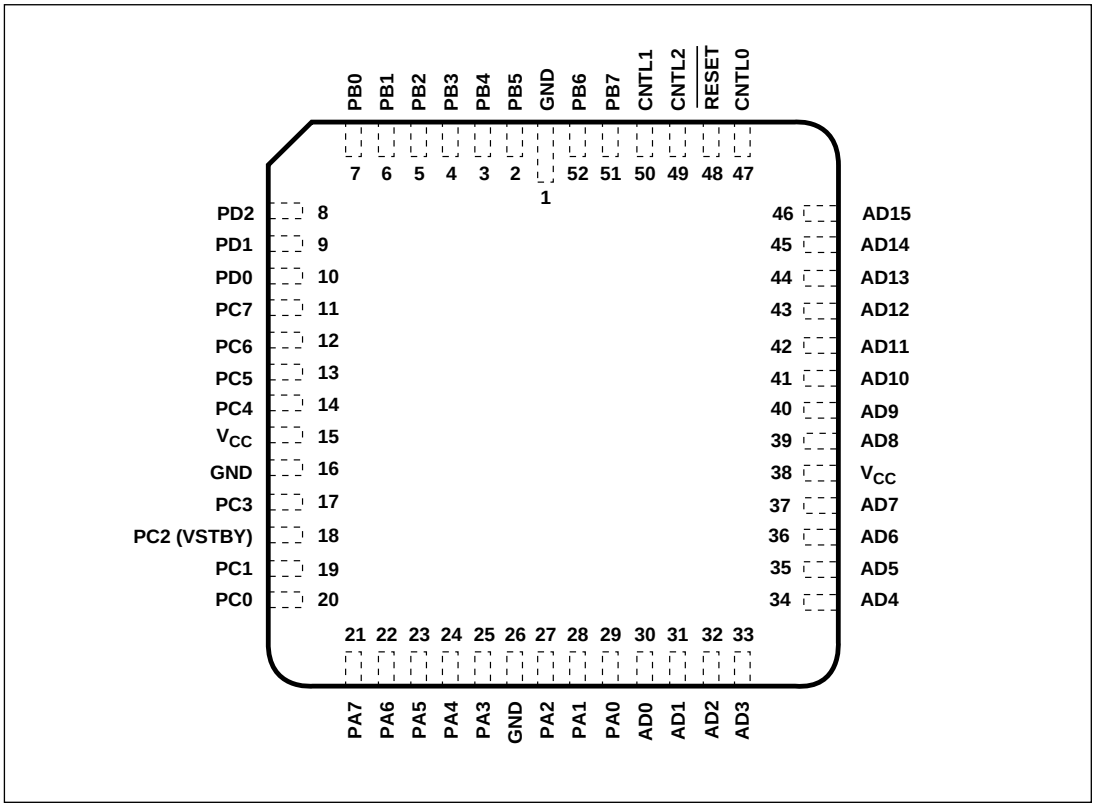



Figure 36. Drawing M3 – 52-Pin Plastic Quad Flatpack (PQFP) (Package Type M)

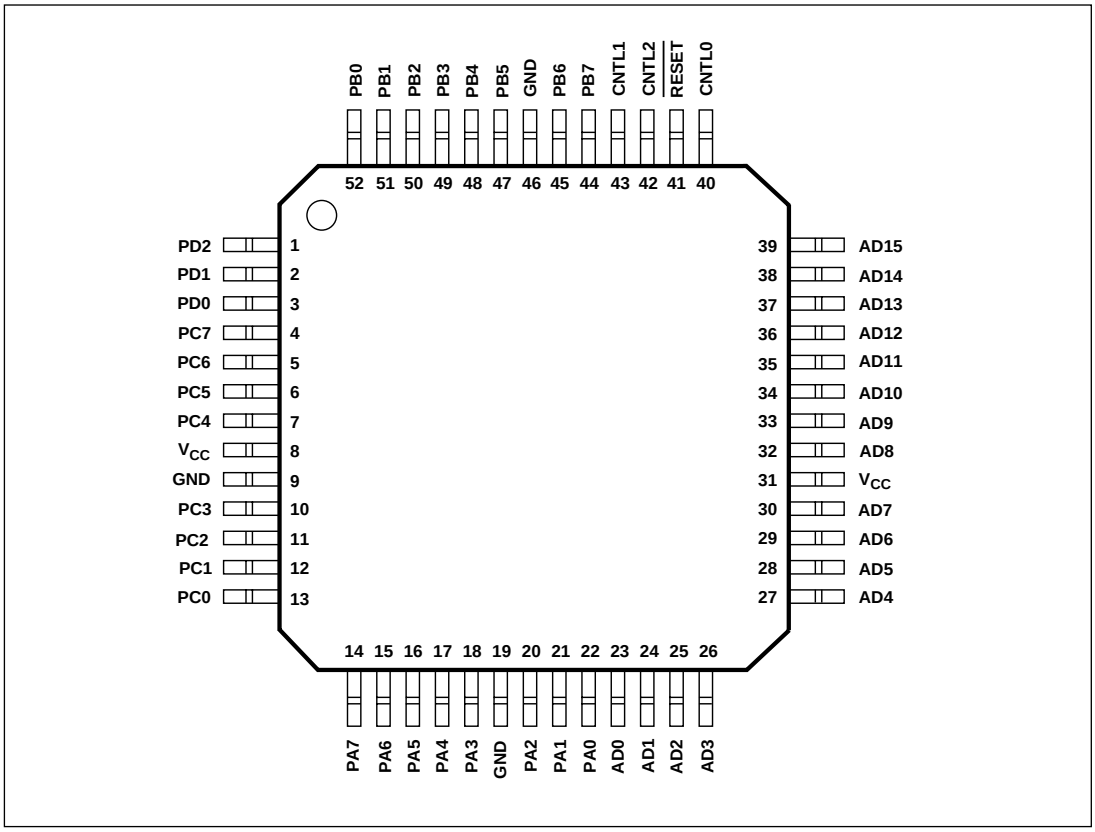
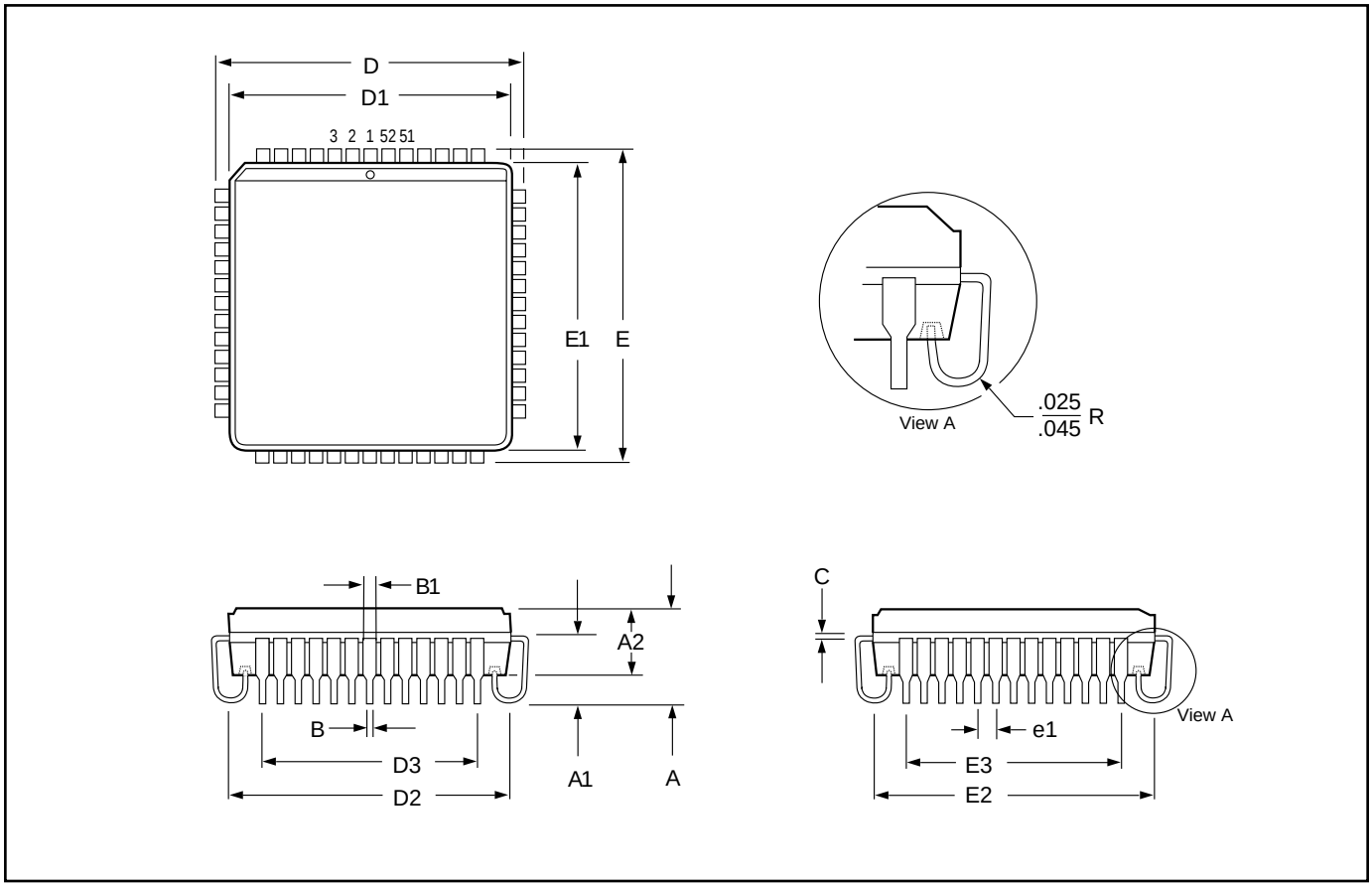


Figure 35A.
Drawing J7 – 52-Pin Plastic Leaded Chip Carrier (PLCC) (Package Type J)

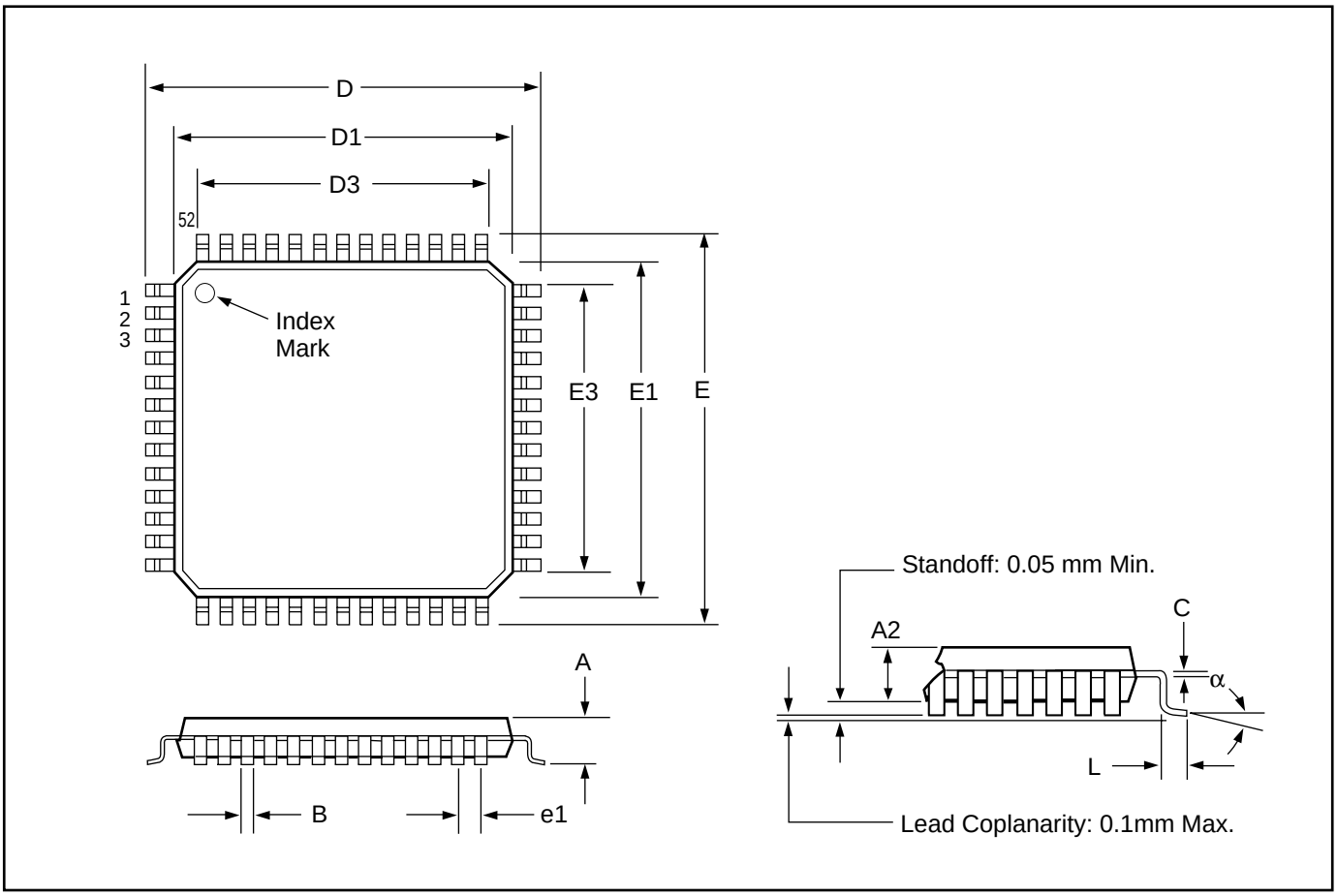



Family: Plastic Leaded Chip Carrier

| | Millimeters | | | Inches | | |
|---|---|---|---|---|---|---|
| Symbol | Min | Max | Notes | Min | Max | Notes |
| A | 4.19 | 4.57 | | 0.165 | 0.180 | |
| A1 | 2.54 | 2.79 | | 0.100 | 0.110 | |
| A2 | 3.66 | 3.86 | | 0.144 | 0.152 | |
| B | 0.33 | 0.53 | | 0.013 | 0.021 | |
| B1 | 0.66 | 0.81 | | 0.026 | 0.032 | |
| C | 0.246 | 0.261 | | 0.0097 | 0.0103 | |
| D | 19.94 | 20.19 | | 0.785 | 0.795 | |
| D1 | 19.05 | 19.15 | | 0.750 | 0.754 | |
| D2 | 17.53 | 18.54 | | 0.690 | 0.730 | |
| D3 | 15.24 | | Reference | 0.600 | | Reference |
| E | 19.94 | 20.19 | | 0.785 | 0.795 | |
| E1 | 19.05 | 19.15 | | 0.750 | 0.754 | |
| E2 | 17.53 | 18.54 | | 0.690 | 0.730 | |
| E3 | 15.24 | | Reference | 0.600 | | Reference |
| e1 | 1.27 | | Reference | 0.050 | | Reference |
| N | 52 | | | 52 | | |

020197R1

Figure 36A.
Drawing M3 – 52-Pin Plastic Quad Flatpack (PQFP) (Package Type M)




Family: Plastic Quad Flatpack (PQFP)

| | Millimeters | | | Inches | | |
|---|---|---|---|---|---|---|
| Symbol | Min | Max | Notes | Min | Max | Notes |
| α | 0° | 7° | | 0° | 7° | |
| A | – | 2.35 | | – | 0.093 | |
| A2 | 1.95 | 2.10 | | 0.077 | 0.083 | |
| B | 0.22 | 0.38 | Reference | 0.009 | 0.015 | |
| C | | 0.23 | | | 0.009 | |
| D | 12.95 | 13.45 | | 0.510 | 0.530 | |
| D1 | 9.90 | 10.10 | | 0.390 | 0.398 | |
| D3 | 7.80 | | Reference | 0.307 | | Reference |
| E | 12.95 | 13.45 | | 0.510 | 0.530 | |
| E1 | 9.90 | 10.10 | | 0.390 | 0.398 | |
| E3 | 7.80 | | Reference | 0.307 | | Reference |
| e1 | 0.65 | | Reference | 0.026 | | Reference |
| L | 0.73 | 1.03 | | 0.029 | 0.041 | |
| N | 52 | | | 52 | | |

060198R0

## Selector Guide

### Selector Guide – PSD9XXF Family

| Part # | | MCU | PLDs/Decoders | | | | | I/O | Memory | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PSD @ 5 V | PSD @ 3 V | Data Path | PLD Inputs | Input Macrocells | Output Macrocells | PLD Outputs | Page Reg. | Ports | Main Flash | Boot Flash | SRAM |
| PSD913F1 | PSD913F1V | 9 | 57 | | | 19 | 8-Bit | 27 | 1024Kb | | 16Kb |
| PSD913F2 | PSD913F2V | 9 | 57 | | | 19 | 8-Bit | 27 | 1024Kb | 256Kb | 16Kb |
| PSD934F2 | PSD934F2V | 9 | 57 | | | 19 | 8-Bit | 27 | 2048Kb | 256Kb | 64Kb |
| PSD954F2 | PSD954F2V | 9 | 57 | | | 19 | 8-Bit | 27 | 2048Kb | 256Kb | 256Kb |

Legend:
PSDV = Zero Power version available at 2.7 V to 5.5 V $V_{CC}$.

## Part Number Construction

**Flash PSD Part Number Construction**

| CHARACTER # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PART NUMBER | P | S | D | 8 | 1 | 3 | F | 2 | | – | A | – | 1 | 5 | J | | | | |

**PSD BRAND NAME**
PSD = Standard Low Power Device

**FAMILY/SERIES**
8 = Flash PSD for 8-bit MCUs
9 = Flash PSD for 8-bit MCUs with Combinatorial PLD

**SRAM SIZE**
0 = 0Kb
1 = 16Kb
2 = 32Kb
3 = 64Kb
4 = 128Kb
5 = 256Kb

**NVM SIZE**
1 = 256Kb
2 = 512Kb
3 = 1Mb
4 = 2Mb

**I/O COUNT & OTHER**
F = 27 I/O

**2ND NVM TYPE, SIZE & CONFIGURATION**
1 = EEPROM, 256Kb
2 = FLASH, 256Kb
3 = No 2nd Array

**TEMP RANGE**
"Blank" = 0°C to +70°C (Commercial)
I = –40°C to +85°C (Industrial)

**PACKAGE TYPE**
J = PLCC
U = TQFP (not available on some)
M = PQFP

**SPEED**
- 90 = 90ns
- 15 = 150ns
- 20 = 200ns

**REVISION**
"Blank" = no rev.
- A = Rev. A

**$V_{CC}$ VOLTAGE**
"blank" = 5 Volt
V = 3.0 Volt

## Ordering Information

| Part Number | Speed (ns) | Package Type | Operating Temperature Range |
|---|---|---|---|
| PSD913F2-70J | 70 | 52 Pin PLCC | Comm'l |
| PSD913F2-70M | 70 | 52 Pin PQFP | Comm'l |
| PSD913F2-90J | 90 | 52 Pin PLCC | Comm'l |
| PSD913F2-90M | 90 | 52 Pin PQFP | Comm'l |
| PSD913F2-90JI | 90 | 52 Pin PLCC | Industrial |
| PSD913F2-90MI | 90 | 52 Pin PQFP | Industrial |
| PSD934F2-70J | 70 | 52 Pin PLCC | Comm'l |
| PSD934F2-70M | 70 | 52 Pin PQFP | Comm'l |
| PSD934F2-90J | 90 | 52 Pin PLCC | Comm'l |
| PSD934F2-90M | 90 | 52 Pin PQFP | Comm'l |
| PSD934F2-90JI | 90 | 52 Pin PLCC | Industrial |
| PSD934F2-90MI | 90 | 52 Pin PQFP | Industrial |
| PSD954F2-70J | 70 | 52 Pin PLCC | Comm'l |
| PSD954F2-70M | 70 | 52 Pin PQFP | Comm'l |
| PSD954F2-90JI | 90 | 52 Pin PLCC | Industrial |
| PSD954F2-90MI | 90 | 52 Pin PQFP | Industrial |
| PSD913F2V-15J | 150 | 52 Pin PLCC | Comm'l |
| PSD913F2V-15M | 150 | 52 Pin PQFP | Comm'l |
| PSD913F2V-20JI | 200 | 52 Pin PLCC | Industrial |
| PSD913F2V-20MI | 200 | 52 Pin PQFP | Industrial |
| PSD934F2V-15J | 150 | 52 Pin PLCC | Comm'l |
| PSD934F2V-15M | 150 | 52 Pin PQFP | Comm'l |
| PSD934F2V-20JI | 200 | 52 Pin PLCC | Industrial |
| PSD934F2V-20MI | 200 | 52 Pin PQFP | Industrial |
| PSD954F2V-90J | 90 | 52 Pin PLCC | Comm'l |
| PSD954F2V-90M | 90 | 52 Pin PQFP | Comm'l |
| PSD954F2V-12JI | 120 | 52 Pin PLCC | Industrial |
| PSD954F2V-12MI | 120 | 52 Pin PQFP | Industrial |

## REVISION HISTORY

**Table 1. Document Revision History**

| Date | Rev. | Description of Revision |
|---|---|---|
| Dec-1999 | 1.0 | Document written in the WSI format |
| Jun-2000 | 1.1 | 3V devices added |
| Nov-2000 | 1.2 | PSD954F2 added |
| 04-Jan-2002 | 1.1 | Front page, and back two pages, in ST format, added to the PDF file<br>References to Waferscale, WSI, EasyFLASH and PSDsoft 2000 updated to ST, ST, Flash+PSD and PSDsoft Express |

Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

The ST logo is registered trademark of STMicroelectronics

All other names are the property of their respective owners

© 2002 STMicroelectronics - All Rights Reserved

STMicroelectronics group of companies Australia - Brazil - Canada - China - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States.

**www.st.com**

WWW.ALLDATASHEET.COM

Copyright © Each Manufacturing Company.

All Datasheets cannot be modified without permission.

This datasheet has been download from :

www.AllDataSheet.com

100% Free DataSheet Search Site.

Free Download.

No Register.

Fast Search System.

www.AllDataSheet.com